**TOURBOT**

**(A CHATBOT USING RASA FOR TRAVEL PLANNING)**

*A project report submitted to ICT Academy of Kerala*

*in partial fulfillment of the requirements*

*for the certification of*

**CERTIFIED SPECIALIST**

**IN**

**Machine Learning and Artificial Intelligence**

submitted by

**Jyolsna K**

**Manu Davis**

**Presi S Premachandran**

**Swathi Rajesh**

**Syam Krishna S**



**ICT ACADEMY OF KERALA**

**THIRUVANANTHAPURAM, KERALA, INDIA**

**Dec 2022**

**List of Figures**

4.3.1 Initializing the model using rasa init command…….19

4.3.2 Folders and files created after run rasa init ……… 20

4.3.2 Chatbot from terminal after rasa shell command…...21

5.1 Result of Tourbot from terminal ……………………24

5.2(a) Result of Tourbot from WhatsApp …………………26

5.3(b) Result of Tourbot from WhatsApp …………………27

5.4(c) Result of Tourbot from WhatsApp …………………28

**List of Abbreviations**

1. AI - Artificial intelligence
2. NLU - Natural language processing
3. BPO - Business Process Outsourcing
4. CI/CD - Continuous Integration and Continuous Deployment
5. DIET - Dual Intent and Entity Transformer
6. REST - Representational State Transfer
7. ML - Machine Learning
8. BERT - Bidirectional Encoder Representations from Transformers
9. ALICE - Artificial Linguistic Internet Computer Entity

**TABLE OF CONTENTS**

**1. PROBLEM STATEMENT 6** 1.1 OVERVIEW ….………….……………….…………… 6

1.2 PROBLEM STATEMENT ..…………..………..……... 7

**2. INTRODUCTION 8**

**3. LITERATURE SURVEY 11**

3.1 HISTORY OF TOUR PLANNING CHATBOTS ………... ..12

**4. METHODOLOGY 14**

4.1 PROPOSED METHODOLOGY...………….….………….…14

4.2 TECHNOLOGIES USED ……...…………….……….……. 14

4.2.1 RASA ……………………….….……………..……. 15

4.2.1.1 RASA NLU……………………………15

4.2.1.2 RASA CORE…………………………..16

4.2.2 NGROK………………………………………………16

4.2.3 TWILIO WHATSAPP API ………………………… 17

4.3 WORKING ……..……………….………………...………... 18

**5. RESULTS 24**

**6. CONCLUSION 29**

**7. REFERENCE 30**

**ABSTRACT**

The project “TOURBOT” aims to develop a tour planning chatbot for the tourism and travel industry using the Rasa framework. The chatbot will be designed to assist travelers in planning and booking their trips, by providing personalized recommendations and assistance with booking flights, hotels, and activities.

The chatbot will utilize machine learning algorithms to understand the preferences and interests of each individual user, and to provide tailored recommendations based on their specific needs and goals. It will also provide information on destinations, attractions, and local events, helping travelers make informed decisions about their travel plans. In addition to providing recommendations and assistance with booking, the chatbot will also help travelers stay organized and on track during their trip, by providing reminders and alerts for flights, activities, and other important events. To develop the chatbot, a dataset of example conversations between the chatbot and users will be collected and used to train the chatbot using the Rasa framework. The chatbot's performance will be evaluated using a separate evaluation dataset, and any necessary adjustments will be made to improve its accuracy and effectiveness.

Overall, the project aims to create a convenient and personalized tool for travelers to use when planning and booking their trips, utilizing the capabilities of the Rasa framework to create a highly effective and efficient chatbot.

**1. PROBLEM DEFINITION**

**1.1 OVERVIEW**

The growth of technologies like Artificial Intelligence has marked many advancements in the technological world. These technologies have been used to solve problems and make our lives easier. Chatbots are one of them. Chatbots are conversational AIs, which mimic the human while conversing and eliminate the need of humans by automating mundane. Chatbots have been found to be useful in customer service because they can answer questions 24/7 without any breaks or holiday's and hence have become a key revolution in almost every sector of industry, ranging from medical to support.

Chatbots act as personal travel assistants to help customers browse flights and hotels, provide budget-based options for travel, and introduce packages and campaigns according to consumers’ travel behavior. Friendly chatbots ensure a better customer experience, thus creating better opportunities for businesses to generate leads through chatbots. Chatbots provide instant response 24/7, across different communication channels, with the ability to hand over conversations to a live agent if needed. The reliability of a chatbot is directly linked to its ability to provide the correct response within a conversation. Reliability depends heavily on the chatbot’s natural language understanding capabilities and data on which it is trained. Having an up-to-date training dataset about travel prices, destinations, packages, and comparisons, ensures travel chatbot reliability.

Through our project "Tourbot", we aim to explore the potential of chatbots in the field of tourism and travel. Tourbot is a web-based application that analyzes and understands user's queries and provides an instant and accurate response. Rasa technology is used to construct this chatbot. It's an open-source technology, which uses its two main packages i.e., Rasa Core & Rasa Natural Language Understanding (NLU) in order to build a Contextual Al Chatbot. NLU is used to infer the intent and to extract the necessary entities from user input & the Rasa Core provides the output by building a probabilistic model.

**1.2 PROBLEM STATEMENT**

As with other businesses, tourist companies are taking advantage of modern technologies. Chatbots are a recent technology that hotels, travel agencies, and airline companies are adopting. Tourists have to spend a lot of time searching for an appropriate flight, hotel or restaurant. They face problems due to scattered data on websites and unable to find user-friendly interfaces. Being in a foreign land, they feel helpless in case of emergency or a panic situation. Their travel experience becomes miserable if they don't have anyone to guide them like a friend.

So our idea is to create a chatbot to help our customers with what they need. The substantial increase in communication on popular instant messaging platforms such as WhatsApp, Facebook Messenger, Snapchat, and Skype have clear implications for companies. Especially in the tourism sector, companies should seize this opportunity to improve their existing services through chat robots, or chatbots. The recognition of the importance of chatbots has continued to grow.

The objective of this project is to develop a tour planning chatbot using the Rasa framework that can assist tourists in planning their tours and vacations. The chatbot should be able to provide information about various tourist destinations, recommend suitable tour packages, assist with bookings and reservations, and provide personalized recommendations based on the user's preferences and interests.

A vast number of chatbots are being built to utilize decision-tree logic, so the response a bot gives depends on keywords identified in the user’s input. The bot takes the user through a conversation route based on what he or she has asked.Only an intelligent chatbot understands and replies to the input query in such a way that users cannot tell a robot from a real customer service operator. By integrating chatbot with WhatsApp can personalize the booking experience, boost customer engagement, and as a result, ensure that your travel and tourism business provides excellent customer service and as a result thrives in the competitive industry.

**2. INTRODUCTION**

In the advent of technology and the digital world, engaging with customers proactively has become a prime goal for enterprises for improving customer service & support, marketing, sales, and generating leads. Gone are the days where all these tasks were outsourced by enterprises to BPOs and Call Centres; not only due to the increased amount of overhead cost but also due to the working hours of these BPOs and Call Centres that used to hinder the process as a whole.

This is where the role of Conversational AI & Chatbot comes into play. Chatbots are software programs with hardcoded logic (possibly if/then statements) with no capacity for learning.A chatbot is often described as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP). Formulating responses to questions in natural language is one of the most typical Examples of Natural Language Processing applied in various enterprises’ end-use applications. Chatbots don’t understand sentences. Instead, they look for specific words that the customer types, and then provide an automated response to those words.

This project deals with building a chatbot in tourism activities. Since chatbots are new age interface that helps to interact with customers, where the interaction is both human-like and personalized, the online assistance which is much of a requirement for the tourists who are planning their holidays for a relatively new place, becomes easier. Travel chatbot helps them figure out what is the greatest deal according to their preferences.

When a tourism industry opts for a travel bot, it eliminates the need for a dedicated team for customer support, which decreases the company expense and team management. Travel and hospitality chatbot pipelined with FAQ database will address most of the queries instantly and there will be less human intervention needed. This results in the need of fewer employees and centralized management. Real-time assistance and location-based suggestions are also add-ons. Today's hospitality customers are more online than offline, comparing the different options available and making bookings online and that is the relevance of such a creation.

This project is about building a chatbot using the RASA framework for tour planning and we have named it TOURBOT. Through our project "Tourbot", we aim to explore the potential of chatbots in the field . Tourbot is a web-based application that analyzes and understands user's queries and provides an instant and accurate response. In our Tourbot, We are focusing on implementing certain features like Tour guide, Hotel enquiry/booking, Travel package , Indian food, Famous tourist spots.

TOURBOT is constructed using RASA framework.It's an open-source technology, which uses its two main packages i.e., Rasa Core & Rasa Natural Language Understanding (NLU) in order to build a Contextual Al Chatbot. NLU is used to infer the intent and to extract the necessary entities from user input & the Rasa Core provides the output by building a probabilistic model. Rasa is an open-source framework to build text and voice-based chatbots. It's working at Level 3 of conversational AI, where the bot can understand the context. A level 3 conversational agent can handle things like the user changing their mind, handling context and even unexpected queries.Rasa is not the only tool available to build a chatbot, but it's one of the best. One thing that is best about RASA is that it follows a modular, extensible, microservices architecture that fits well in a typical software development scenario. RASA also provides an easy setup with CI/CD Deployment tools like GitHub Actions when taking the bot to production.RASA provides easy integration capabilities using REST API channels with popular messaging services like WhatsApp, Slack, Telegram, Facebook Messenger, Rocket.Chat etc Other than that, one can also host the chatbot on their dedicated website easily.

RASA uses DIET (Dual Intent and Entity Transformer) as part of its NLU architecture. DIET is a multi-task transformer architecture that handles both intent classification and entity recognition together. DIET improves upon current state-of-the-art architectures and is six times faster to train along with the ability to plug-and-play various pre-trained embeddings like BERT. It also has support for custom components and pipelines to use any other ML model.

Chatbots are one best example that shows technology taking its space in the advanced form in every field of day-to-day life.This project on TOURBOT also looks forward to such an initiative in the field of travel and tourism, as an initial step, and more.

**3. LITERATURE SURVEY**

A chatbot is a computer program designed to simulate conversation with human users, especially over the Internet. Chatbots are often used as a customer service tool, as they can provide quick and accurate responses to a wide range of inquiries and requests. In the travel industry, chatbots have become a popular way for companies to interact with customers and provide information about travel plans, destinations, and services. There are many ways that chatbots can be used in the travel industry. They can be used to answer frequently asked questions, such as those about flight times, baggage policies, and hotel amenities. They can also help customers book flights, hotels, and rental cars, and provide recommendations for things to do and see at their destination. Some chatbots even have the ability to understand natural language and provide personalized recommendations based on a customer's interests and preferences. Chatbots have the potential to greatly improve the travel experience for both companies and customers. They can reduce the workload of customer service staff, and provide a convenient and efficient way for customers to get the information they need. As chatbot technology continues to advance, it is likely that they will become an increasingly important part of the travel industry.

The concept of chatbots, or computer programs designed to simulate conversation with human users, has been around for decades. One of the earliest examples of a chatbot was ELIZA, a program developed in the 1960s by Joseph Weizenbaum at MIT. ELIZA was designed to mimic the responses of a psychotherapist and used simple pattern matching to generate responses to user inputs. Since then, chatbots have evolved significantly, with the development of more sophisticated natural language processing algorithms and the ability to access vast amounts of data and information. Chatbots are now used in a variety of applications, including customer service, sales and marketing, and education. They can be accessed through a variety of channels, including websites, mobile apps, and messaging platforms. There have been many different chatbots developed throughout history, with a wide range of purposes and capabilities.

* ELIZA (1960s): Developed at MIT, ELIZA was one of the first chatbots and was designed to mimic the responses of a psychotherapist.
* PARRY (1972): Developed by psychiatrist Kenneth Colby, PARRY was designed to simulate the responses of a paranoid schizophrenic. It was used as a tool for research into the nature of mental illness.
* Jabberwacky (1988): Developed by Richard Wallace, Jabberwacky was one of the first chatbots to use artificial intelligence techniques such as machine learning. It was designed to be able to carry out a conversation with human users in a natural and engaging way.
* SmarterChild (2001): Developed by ActiveBuddy, SmarterChild was a chatbot that could be accessed through instant messaging platforms. It was designed to provide information and assistance to users, such as weather updates and directions.
* Apple's Siri (2011): Siri is a voice-activated virtual assistant developed by Apple for its iOS, iPadOS, watchOS, macOS, and tvOS operating systems. It uses artificial intelligence and natural language processing to understand and respond to user requests and provide information and assistance.
* Microsoft's Cortana (2014): Cortana is a virtual assistant developed by Microsoft for its Windows operating system. Like Siri, it uses artificial intelligence and natural language processing to understand and respond to user requests and provide information and assistance.

**3.1 HISTORY OF TOUR PLANNING CHATBOTS**

The use of chatbots for tourism purposes is a relatively recent development, and has been made possible by advances in artificial intelligence and natural language processing. One of the earliest examples of a tourism chatbot was Expedia's Expedia Assist, which was launched in 2016. Since then, a number of other companies in the travel industry, including airlines, hotel chains, and online travel agencies, have developed chatbots to assist users with booking and planning travel arrangements. Tourism chatbots have generally been well-received by users, as they provide a convenient and efficient way to access information and make travel arrangements. However, they may not always be able to fully replace human travel agents, as they may not be able to handle more complex or nuanced requests, or provide personalized recommendations based on a deep understanding of a user's preferences and needs.

* Expedia: Expedia's chatbot, called Expedia Assist, allows users to book flights, hotels, and rental cars through a conversational interface. It can provide price quotes, availability information, and help users compare options. Expedia Assist is available through the Expedia website and mobile app, as well as through messaging platforms such as Facebook Messenger and WhatsApp.
* KLM: KLM, the Dutch airline, has a chatbot called KLM Royal Dutch Airlines Bot that allows users to book flights, check flight status, and receive notifications about their travel plans. The chatbot is available through messaging platforms such as Facebook Messenger, WhatsApp, and WeChat.
* TripAdvisor: TripAdvisor's chatbot, called the TripAdvisor Chatbot, provides information about destinations, hotels, restaurants, and attractions, and can recommend places to visit based on a user's interests and preferences. It is available through messaging platforms such as Facebook Messenger and WhatsApp.
* Kayak: Kayak's chatbot, called the Kayak Chatbot, allows users to search for and book travel arrangements, such as flights, hotels, and rental cars. It can provide price quotes and availability information, and help users compare options. The Kayak Chatbot is available through messaging platforms such as Facebook Messenger, Skype, and Slack.
* Travelocity: Travelocity's chatbot, called the Travelocity Chatbot, allows users to book flights, hotels, and rental cars through a conversational interface. It can provide price quotes, availability information, and help users compare options. The Travelocity Chatbot is available through messaging platforms such as Facebook Messenger, WhatsApp, and WeChat.

**4. METHODOLOGY**

**4.1 PROPOSED METHODOLOGY**

The project aims to develop a tour planning chatbot for the tourism industry using the Rasa framework. The chatbot will be designed to assist travelers in planning and booking their trips, by providing personalized recommendations and assistance.

We will use the rasa framework to build the required chatbot and we are integrating this chatbot to WhatsApp. Below are the key points that we considered for choosing the RASA framework for creating chatbot.

* Highly customizable with various pipelines can be employed to process user dialogues.
* The rasa framework can be run as a simple http server or can be used from python, using APIs.
* It has the RASA-NLU, when run on a server, can mimic other commercial NLP platforms such as LUIS or wit.ai.

**4.2 TECHNOLOGIES USED**

Here are just a few instances of technologies used in our project:

1. RASA
2. Ngrok
3. Twilio WhatsApp APIs

**4.2.1 RASA**

The Rasa framework is an open-source platform for building chatbots and conversational AI systems. It was designed to be flexible and customizable, and can be used to build a wide range of chatbot applications. One of the key capabilities of the Rasa framework is its natural language processing (NLP) capabilities. Rasa uses machine learning algorithms to understand and interpret human language, allowing it to respond to user inputs in a way that feels natural and human-like. Rasa also has the ability to learn from user interactions and improve its responses over time, making it a powerful tool for building chatbots that can handle complex conversations.

RASA, an open-source implementation, uses the Dual Intent and Entity Transformer (DIET) model of natural language processing (NLP). RASA is made up of two modules: Rasa NLU and Rasa Core. Input from the user is examined by Rasa NLU, which then categorizes the intent and extracts the entities. Rasa core receives user input and uses a variety of pipelines to produce an appropriate response. Rasa is a powerful and quick tool for creating sophisticated chatbots that manage dialogue right out of the box. In terms of development, it is open and scalable.

**4.2.1.1 RASA NLU**

If user data contains entities, those entities must be extracted. A dictionary containing the original text, the intended meaning, and any entities found are extracted from the input message by the RASA framework's interpreter and sent to the rasa core (Dialogue management) for further processing. Tokenization of the text (i.e., splitting each word into a token), featurization (numeric feature) for the tokenized input text, intent classification, and entity extraction (if necessary) are all handled by default rasa pipelines or custom pipelines in this NLU process.

**4.2.1.2 RASA CORE**

Rasa Core is responsible for handling context handling, bot responses, and session management. It is built on top of Rasa NLU (Natural Language Understanding), which is responsible for understanding the meaning and intent of user input. Rasa Core uses this understanding of user intent to decide what action to take next in the conversation, and to keep track of the context and state of the conversation. Rasa core receives structured data from the interpreter in the form of intents and entities, which it uses to decide which action to take for a given input using a probabilistic model. An action is something that is performed, like showing something to the user or sending them a message. Rasa Core's decision is based on entities and intents that have been trained using machine learning models.

Rasa core to carry out actions specific to the input through an API call. We have multiple opportunities to influence these processes. First, the interpreter, or Rasa NLU, is set up. This ought to be able to reliably identify the proper intents and extract all required entities. The Policy component can also be customized for each Use Case. We can further customize our settings by selecting one of the many policies available to us.

The activity is Bot Responses. Responses, Quick Replies, images, and action webhooks can all be delivered using plain text in this way. The latter sends a POST request to a pre-set interface, from which it receives replies.

This enables API calls as well as database searches. So, thecombination of Rasa NLU and Rasa core is termed as Rasa stack. In addition to managing the conversation flow, Rasa Core also includes a number of other useful features for chatbot development. It allows developers to build custom actions and logic into chatbot conversations, and to integrate with external APIs and databases to access and retrieve information. It also has support for a variety of messaging platforms, making it easy to build chatbots that can be used on a wide range of devices.

**4.2.2 NGROK**

Ngrok is a tool that allows you to expose a local web server to the internet, by creating a secure tunnel between your local machine and the ngrok cloud service. This allows you to test webhooks, APIs, and other web-based systems that are running on your local machine, even though it's only accessible from localhost.

It creates a unique public URL for your local web server, so you can test it from anywhere, as if it were running on a public domain. It's very handy when you're developing web applications and need to test webhooks or APIs before deploying them to a live environment. Ngrok can be easily downloaded and installed, and can be used with any web server, framework, or development stack.

Ngrok is a tool that allows you to expose a local web server to the internet. It creates a secure tunnel to your localhost and assigns a unique public URL to your development environment, which can be accessed from anywhere on the internet. Ngrok allows you to test and debug webhooks, APIs, and other web-based systems that are running on your lo

**4.2.3 TWILIO WHATSAPP API**

Twilio WhatsApp API is a messaging platform that allows you to send and receive messages through the WhatsApp business application programming interface (API).Using the Twilio WhatsApp API, you can build chatbots and messaging applications that can send and receive messages on WhatsApp. You can use the API to send text messages, images, videos, and other media, as well as to send and receive location information and status updates.

To get started with the Twilio WhatsApp API, you will need to sign up for a Twilio account and purchase a WhatsApp API Sandbox or a WhatsApp API Production plan. You will also need to set up a WhatsApp business account and register your phone number with Twilio. Once you have set up your Twilio account and connected it to your WhatsApp business account, you can use the API to send and receive messages through the WhatsApp API. You can use the API with a variety of programming languages and frameworks, including Python, Java, Node.js, and Ruby.

**4.3 WORKING**

The working of our proposed system can be easily explained by the following steps.

Rasa needs python to work, so Install python version 3.7 to 3.8 at maximum in our system. If Python environment is already configured in the system, we need to check the version by opening the terminal and run the command

* **python ‐‐version**

Version of python is: Python 3.8.15

Create a new folder for chatbot

create a new virtual environment and activate this environment by following steps

* **conda create -n new\_env python=3.8**
* **conda activate new\_env**

Install all necessary packages

* **conda install ujson**
* **conda install tensorflow**

Upgrade the pip to latest form and Install Rasa open-source with the below commands

* **pip3 install -U pip**
* **pip install rasa**

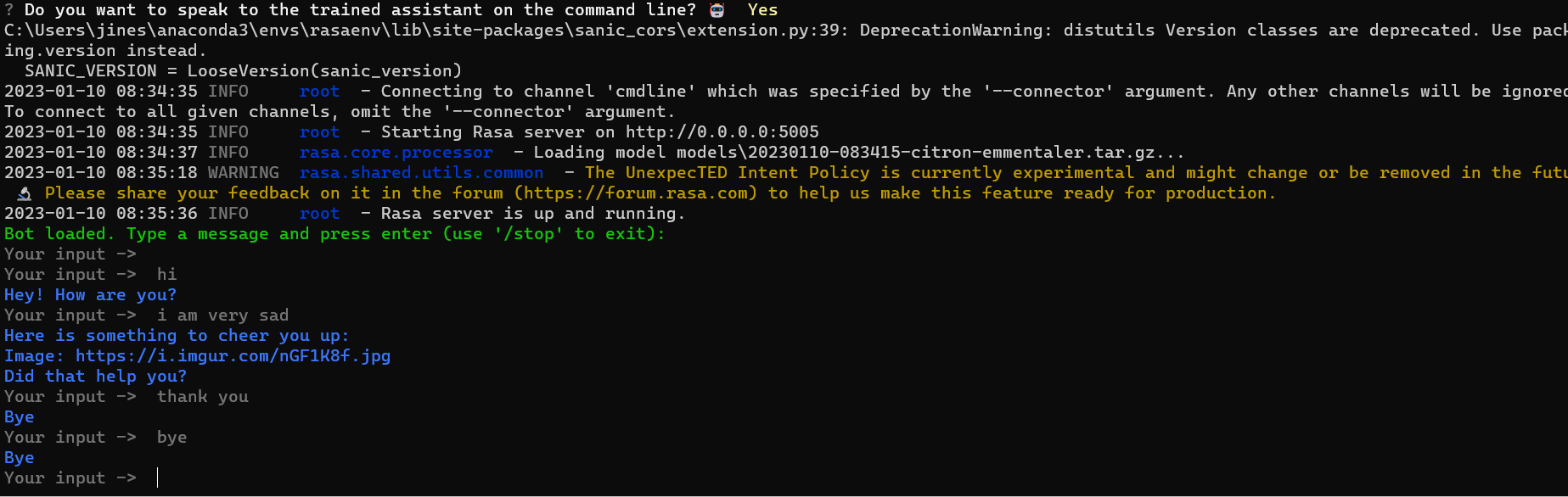
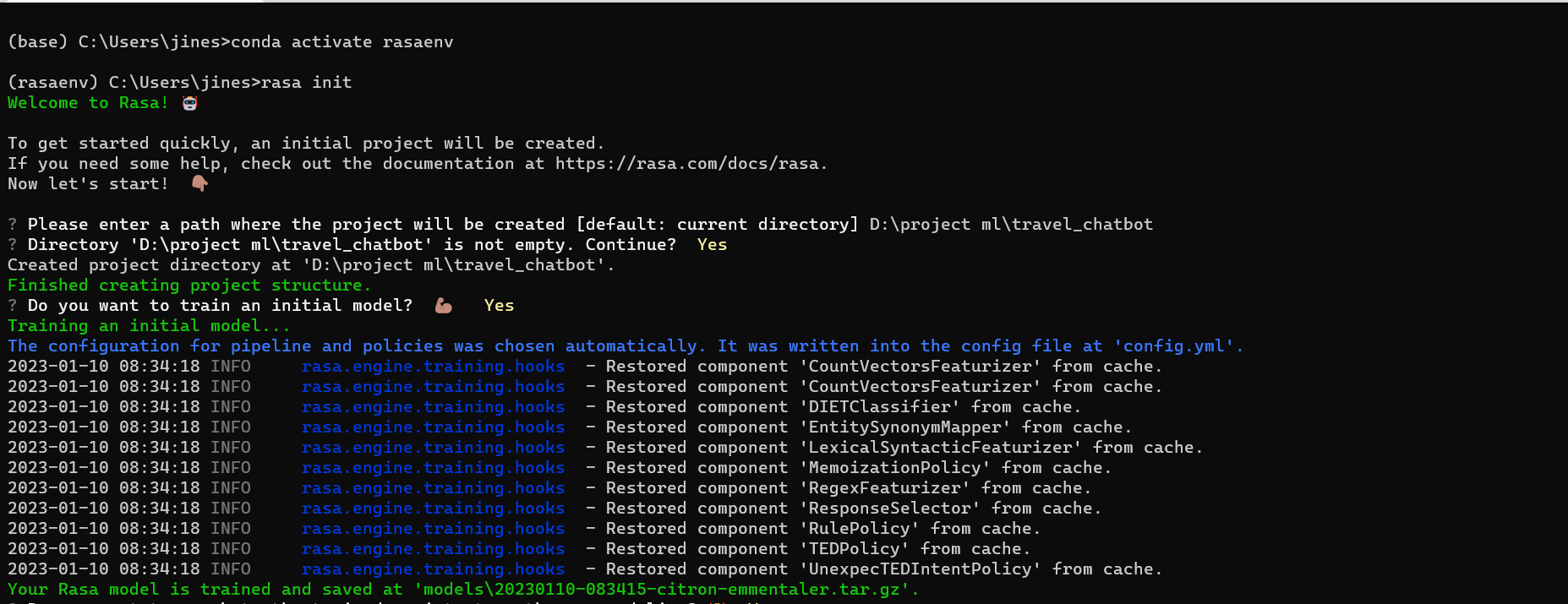
Check the rasa version

* **rasa –version**

Rasa Version : 3.4.0

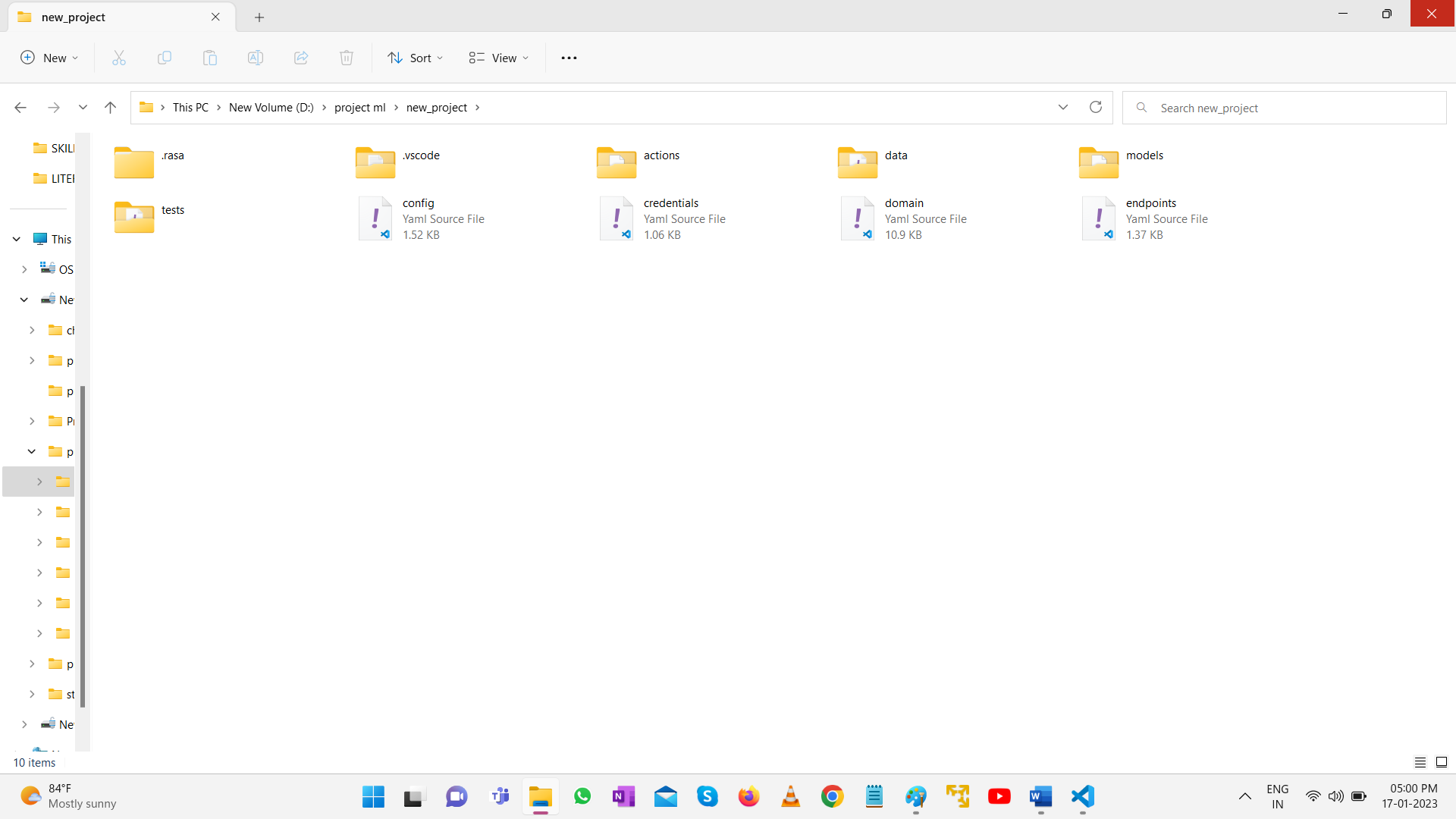
Create the initial demo chatbot with below command

* **rasa init**

****

**Fig 4.3.1 initializing the model using rasa init command**

We can stop the chat by typing the command **/stop** in the terminal.



**Fig 4.3.2 Folders and files created after run rasa init**

Now open the TOURBOT folder in Visual Studio Code .

* **preferences: open workspace settings(JSON)**
* Create a json file VSC

**{**

**"Python. terminal. activateEnvironment”: true,**

**"Python. condapath":"C:\\Users\\jines\\anaconda3\\envs\\new\_env"**

**}**

* **"Terminal: Select Default Profile"**
* Select Command prompt from the dropdown options. (**command prompt: C:\Windows\System32\cmd.exe**)

Now open the terminal in VSC and open and activate environment

* **C:/Users/jines/anaconda3/Scripts/activate**
* **conda activate new\_env**

Make required changes in nlu.yml, rules.yml, stories.yml, domain.yml, credential.yml, action.py files and save it.

Now train the model using the command.

* **rasa train**

new model will be created in model folder.

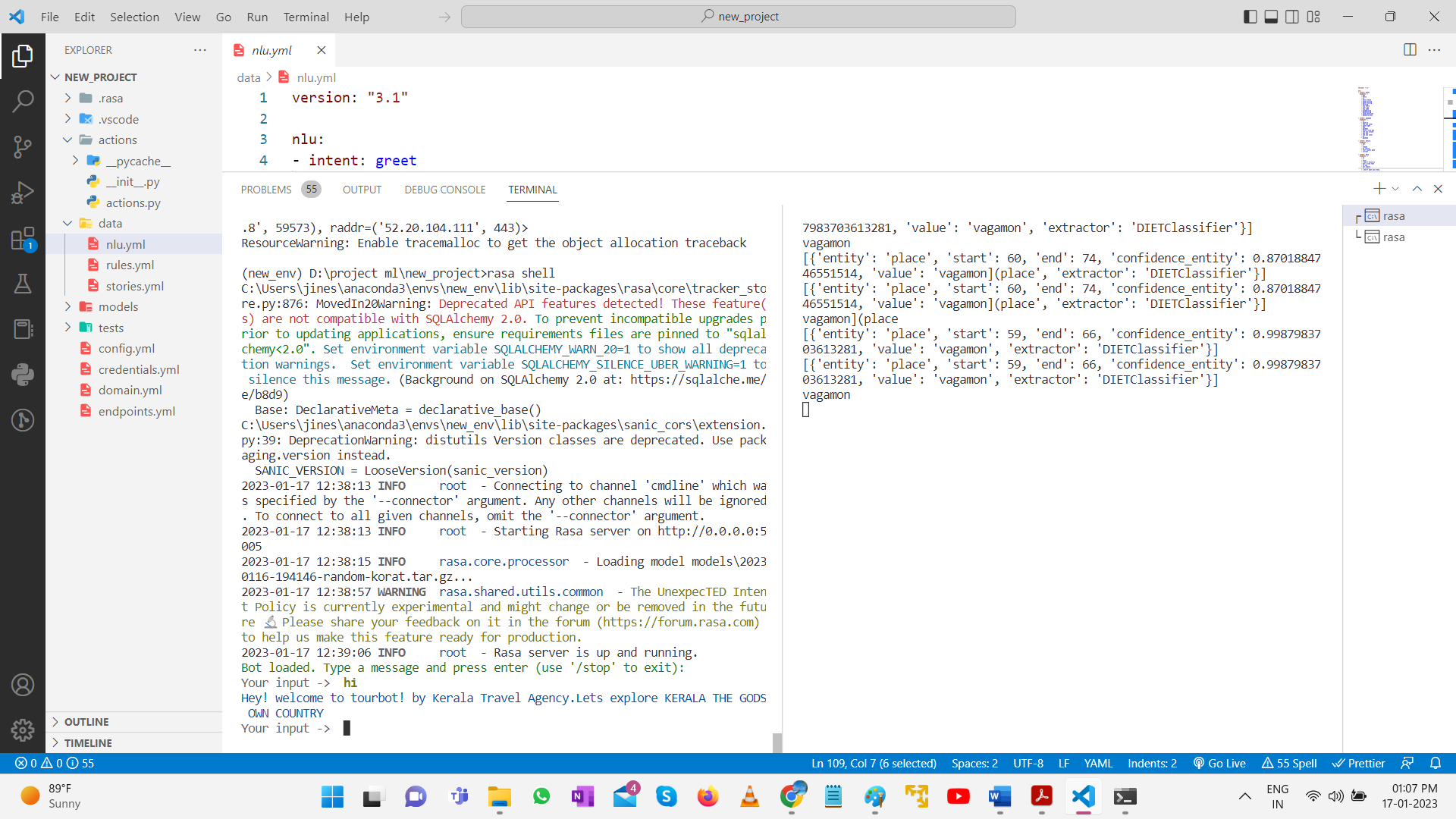
Open another terminal.

To run the actions in action.py file, using the following command in the second terminal

* **rasa run actions**

We can chat with the trained chatbot from the terminal itself, using the command in the first terminal

* **rasa shell**



**Fig 4.3.3 Chatbot from terminal after rasa shell command**

We can stop the chat by typing the command **/stop** in the terminal. Similarly we can use the **rasa run**  command to run the chatbot on the server.

Download and install ngrok . Then extract ngrok from the terminal.Start a tunnel using the command

* **ngrok http 5005**

copy the forwarding url from ngrok and paste it in twilio as

**https://e366-2001-f40-970-2690-6423-5e0e-3402-e724.ap.ngrok.io /webhooks/twilio/webhook**

To connect the Twilio WhatsApp API to a Rasa chatbot, you will need to set up a webhook that allows your chatbot to receive and respond to messages from the WhatsApp API. Here are the steps you can follow to connect the Twilio WhatsApp API to a Rasa chatbot:

1. Set up a Twilio account and purchase a WhatsApp API Sandbox or Production plan.

* Go to the Twilio website ([Twilio](https://www.twilio.com/)) and click on the "Get started for free" button.
* Enter your email address and create a password to sign up for a Twilio account.
* Verify your email address by clicking on the verification link sent to your email.
* After verifying your email, log in to your Twilio account and click on the "Get a number" button in the dashboard.
* Select "WhatsApp" as the number type and choose a number from the list of available numbers.
* Click on the "Buy a number" button to purchase the number.
* After purchasing the number, click on the "Programmable SMS" tab in the dashboard and then click on the "WhatsApp" tab.
* Click on the "Get started" button to purchase a WhatsApp API Sandbox or Production plan.
* Set up a WhatsApp account and register your phone number with Twilio.

1. Set up a Rasa chatbot and define the chatbot's natural language processing (NLP) and dialogue logic.
2. We train our model using the **rasa *train*** command. This will train our NLU model.
3. Configure the Twilio WhatsApp API to send messages to your chatbot via a webhook.

Now, go to the ***endpoints.yml*** file and uncomment the following lines:

* **action\_endpoint:**

**url:** [**http://localhost:5055/webhook**](http://localhost:5055/webhook)

Then, go to the ***credential.yml*** file and type

**twilio:**

**account\_sid: "ACa174ba86df8e6c396268fa75b243b958"**

**auth\_token: "521d08ed9deb1f10313ec317er25691f"**

**twilio\_number: "whatsapp:+14155238886"**

1. Use the Rasa SDK to send the incoming message to the Rasa chatbot backend and receive the chatbot's response and Use the Twilio WhatsApp API to send the chatbot's response back to the user via WhatsApp

Connected it to WhatsApp using twilio by using the command in first terminal:

* **rasa run -m models --enable-api --cors “\*” --debug**

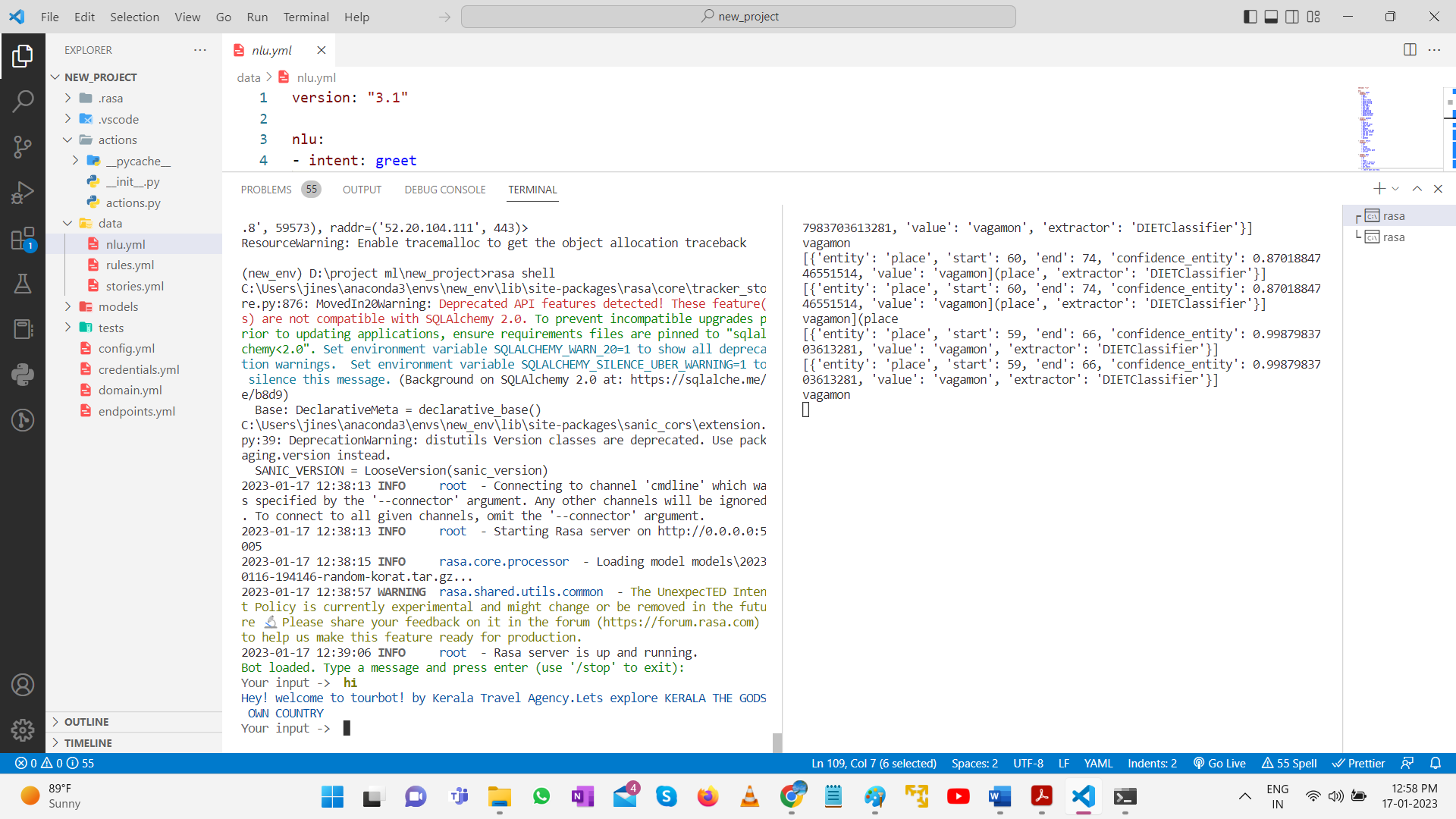
and in the second terminal is

* **rasa run actions**

Hence integrated rasa and WhatsApp.

**5. RESULT**

We could create the TOURBOT using rasa, by making required changes in nlu.yml, rules.yml, stories.yml, domain.yml, credential.yml, action.py files. The model is working fine and is able to give the expected results from the terminal. We also tried to integrate the chatbot which was working fine on the terminal to our whatsapp.



**Fig 5.1 Result of Tourbot from terminal**

The chatbot that we created using rasa and integrated with WhatsApp using the Twilio WhatsApp API has demonstrated several key capabilities:

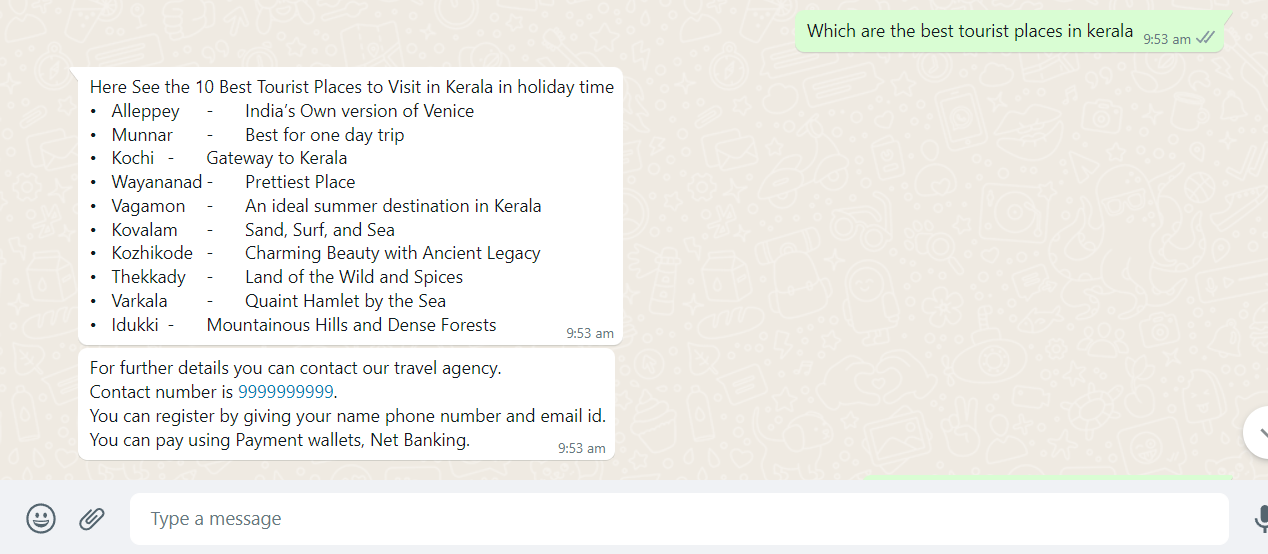
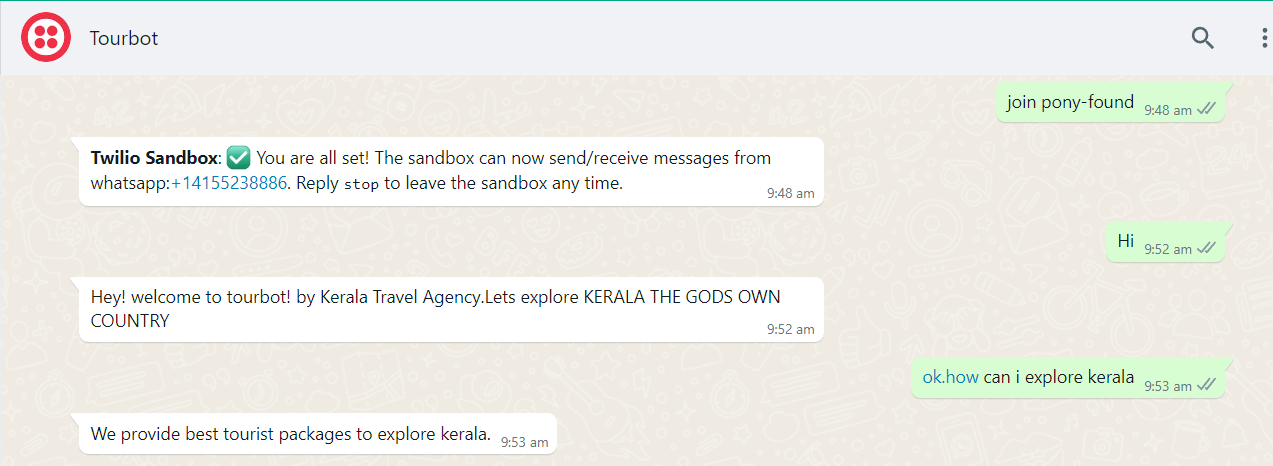
* It can receive and respond to messages from users via WhatsApp.
* It can understand and interpret the user's intentions and provide appropriate responses based on its natural language processing and dialogue logic.
* It can learn from user interactions and improve its performance over time using machine learning algorithms.
* It can communicate with users in a natural and engaging way, providing a personalized and convenient customer experience.

The following are the functions we implemented in our Tourbot which helps in planning our vacation in kerala

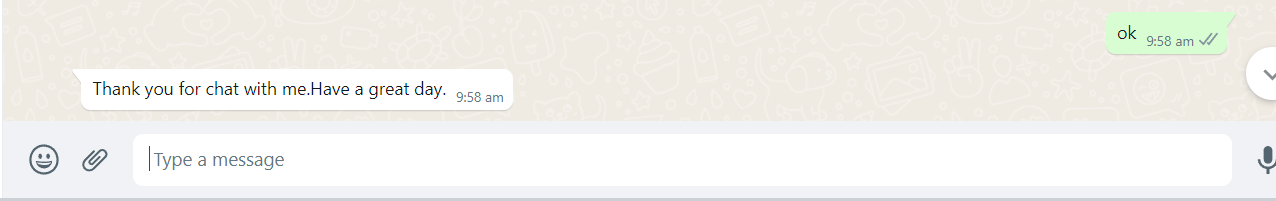
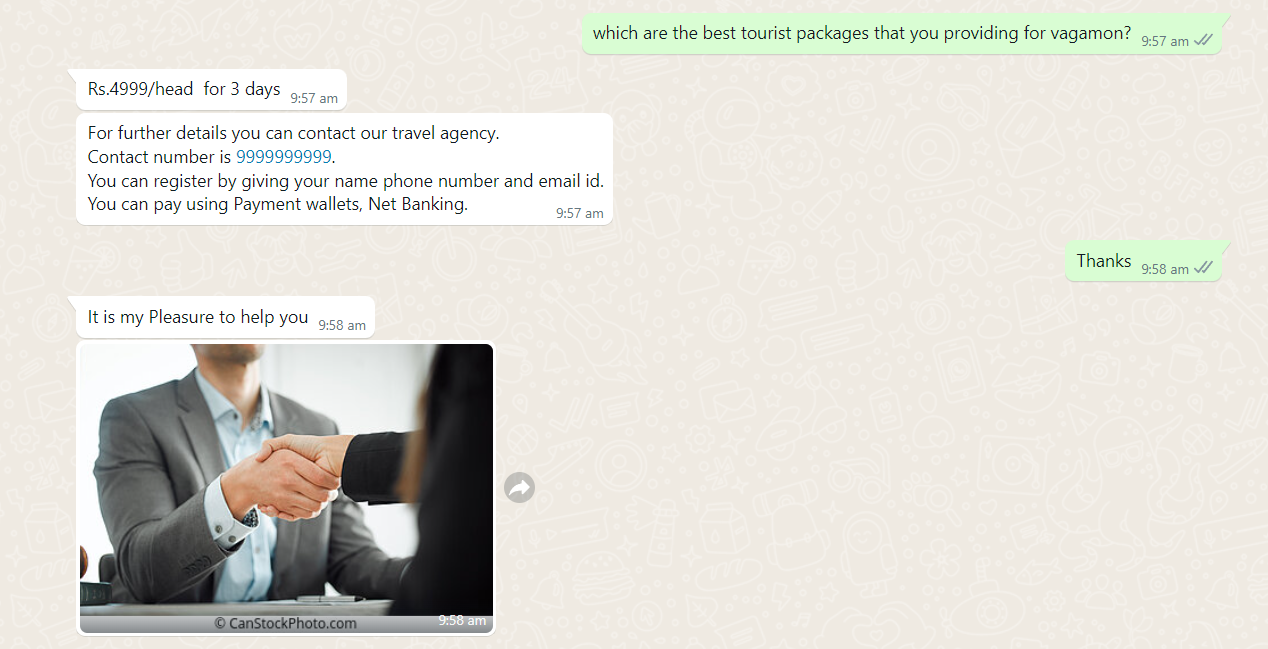
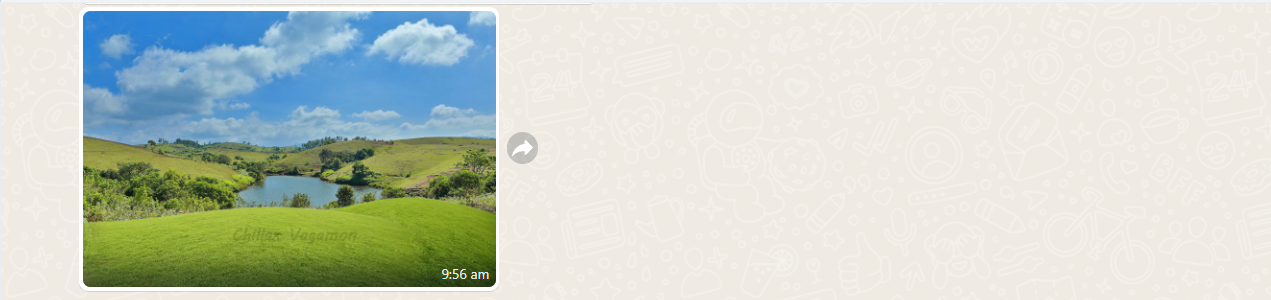
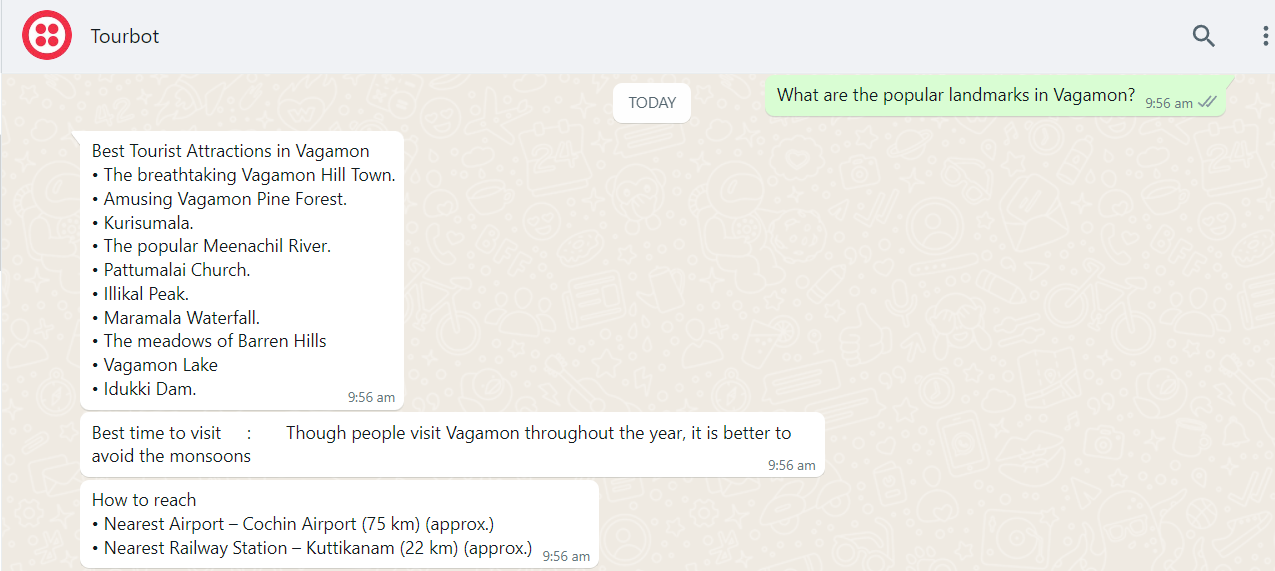
* List top 10 tourist destination in Kerala and greeting the customer
* Description of each destination
* Give recommendation on the time to visit for each destination
* Show the information on how to visit
* Included pictures of each destination

Overall, the chatbot has achieved the goal of creating a powerful and effective communication channel between your business and its customers through WhatsApp. Users can easily access the chatbot anytime, anywhere, and receive timely and relevant responses to their questions and requests. The chatbot has also helped to reduce the workload of customer service staff and improve the efficiency of your business operations.

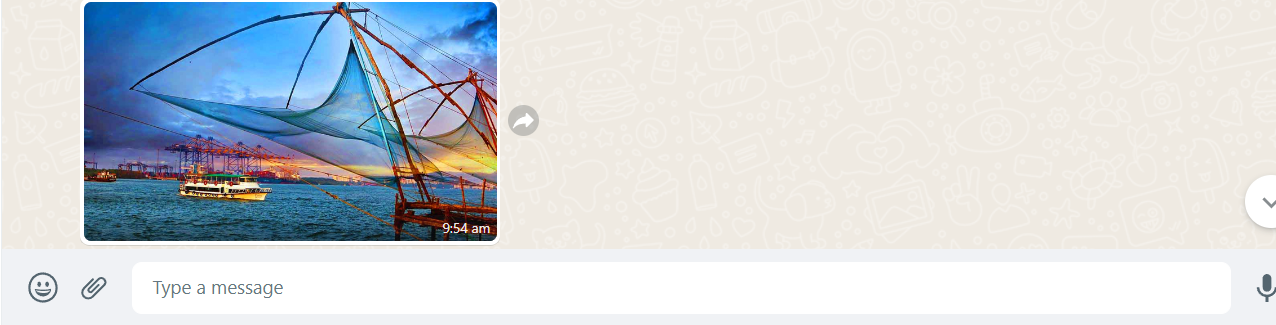
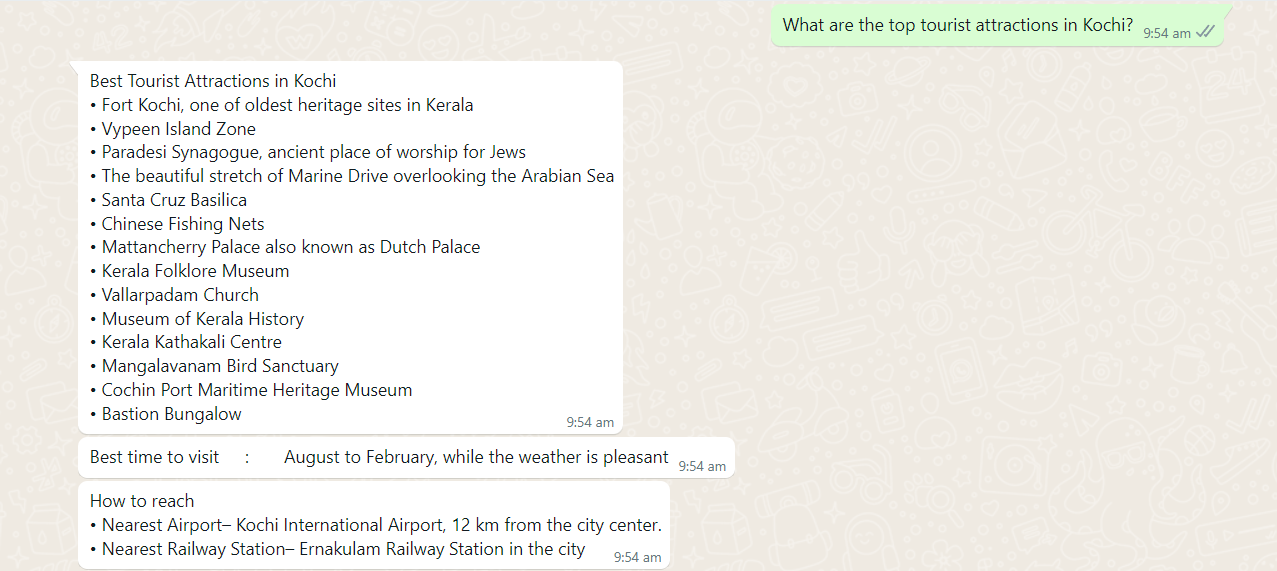
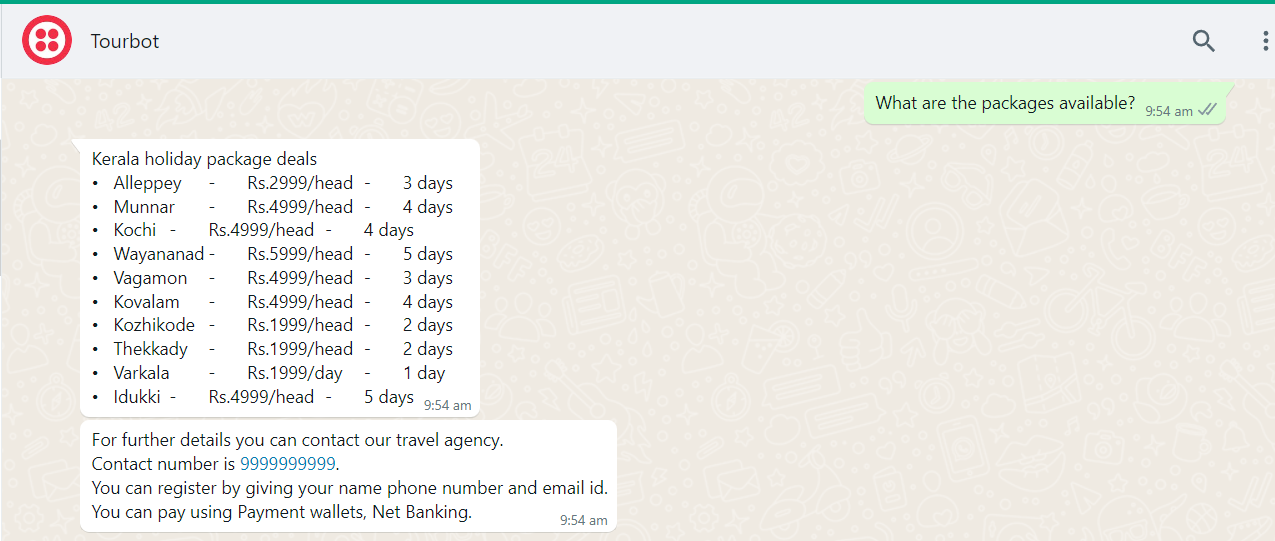
The screenshots of tourbot from WhatsApp are given below.



**Fig 5.2(a) Result of Tourbot from WhatsApp**



**Fig 5.2(b) Result of Tourbot from whatsapp**



**Fig 5.2.(c) Result of Tourbot from WhatsApp**

**GitHub repo link for codes:**

**https://github.com/PRESISPREMACHANDRAN/ML-AI-PROJECT-RASA-CHATBOT-FOR-TOUR-PLANNING**

**6. CONCLUSION**

In conclusion, we have successfully created and integrated a chatbot with WhatsApp using the Twilio WhatsApp API. We have set up a Twilio account, purchased a WhatsApp API plan, set up a WhatsApp business account, and registered your phone number with Twilio. We have also configured the Twilio WhatsApp API to send messages to your chatbot via a webhook. We have tested the chatbot by sending it messages from your WhatsApp business account and observed the chatbot's responses.

Overall, this chatbot project has demonstrated the power of Rasa and the Twilio WhatsApp API in building intelligent and engaging chatbot applications that can be accessed through WhatsApp. The development of a chatbot using the Rasa framework and integrating it with the WhatsApp API through Twilio is a powerful solution for businesses looking to improve their customer service and engagement. The Rasa framework provides a flexible and efficient way to create a chatbot that can understand natural language input and respond in a personalized way. The integration with the WhatsApp API allows the chatbot to reach customers where they are already active and engaged, providing a convenient and seamless experience for the user. The use of webhooks and the Twilio API allows the chatbot to receive real-time information and respond accordingly. The use of ngrok, also allows testing the chatbot locally before deploying it to a live environment. The whole process was cost-effective and easy to implement, providing a scalable solution for businesses of all sizes. Overall, the development of a chatbot using Rasa and integrating it with WhatsApp through Twilio is a valuable investment for businesses looking to improve their customer service and engagement.

**7. REFERENCES**

* <https://www.chatbots.org>
* <https://rasa.com/docs/rasa/installation/installing-rasa-open-source>
* <https://www.geeksforgeeks.org/chatbots-using-python-and-rasa>
* <https://www.twilio.com/> [/WhatsApp](https://www.twilio.com/whatsapp)
* <https://ngrok.com/>