



**I N N O M A T I C S**  
**R E S E A R C H L A B S**

**MAJOR PROJECT REPORT  
ON**

**Covid-19 Chatbot using RASA and NLP**

**SUBMITTED BY**

**Najmush Saquib Ali**  
**Sachin Kumar Gupta**  
**D Sikha Datta Reddy**  
**Shubham Salokhe**

**BATCH**

**OCT -21**

**GUIDED BY**

**Mr. Himanshu Goswami**  
**Mr. Kanav Bansal** (Chief Data Scientist)

## Summary

This project's main aim and objective are to build a Chatbot for questions regarding the coronavirus. Every website uses a Chatbot to interact with the users and help them out. This has proven to reduce the time and resources used to a great extent. The coronavirus outbreak has significant consequences for society worldwide. People are rightly concerned and have many urgent questions. The World Health Organization provides answers to frequently asked questions regarding the coronavirus. However, you may have to search for a while before finding the correct answer to your question. People must be well informed about current measures. This way, we can efficiently limit the mass spread and achieve herd immunity. A chatbot could ideally help with this! This chatbot is developed using RASA (Receive, Appreciate, Summarize, Ask) Framework and NLP (Natural Language Processing}.

## Problem Statement

In the light of the global COVID19 pandemic, it is now doubly important for the governments of various states to increase and inject funding into healthcare and medicine. This will also prompt the demand for technological infrastructure. This is where an automated Chatbot dealing with medical issues can provide respite to many. So, we have to make a Chatbot using Natural Language Processing and Rasa that helps users in issues due to COVID19 by answering questions like:

- Precautionary measures
- Availability of medical health professionals
- Medicines availability
- Basic amenities like food, water, grocery, etc.
- Nearest testing centers, nearby shelter homes
- Statistics based on districts or states in India

Technologies used-

- DL
- NLP
- RASA

## What is RASA –

**Rasa** is a python framework that helps us to build any kind of Chatbot easily. It is based on NLU (Natural Language Processing) which offers the possibility to understand what the user wants.

The Chatbot is composed by some specific terms that we will discover:

- **Intents:** It represents intentions that the user wants to express when he submits his message to the Chatbot. Inside the intents, we provide some phrases that the user may ask and we also provide some responses that the Chatbot must use to answer to the user. It's the work of the Developer here.
- **Entities:** Entities are keywords that represent some specific data that the Chatbot may use to perform the discussion with the user. Entities are used to extract some values inside the user input (message).

Now, we know how Chatbot works, we can switch to the Rasa Framework.

Rasa is split into two python libraries like Rasa NLU and Rasa Core. The first one is the Natural Language Understanding and uses the intents to understand what the user wants and entities to extract some specific values to make the conversation more interesting, after Rasa NLU has understood what the user wants and extracted some values, it sends it to the next one. Rasa Core receives the data sent by Rasa NLU and processes it to find the correct answer that it should send to the user as output, for having a response, it will look for the responses that the Developer provided to him inside the intents.

## Project Files –

**The project consists of the following files:**

1. **nlu.yml**- This file contains the list of intents and their possible sample text. Also, we will map our entities' text. separately in this sample. These intents are used to train our NLU model.
2. **domain.yml**- This file lists all intents, the bot's responses with their text, and also the action that can be performed by the bot. Note, any action or response we define has to be listed in the domain.yml
3. **stories.yml**- This file gives the bot an idea of how the conversation should flow. These stories are defined in the form of markdown language. We delete some predefined stories and replace them with those conversations which are relevant for our bot.
4. **Vaccination file (action file)**
5. **Cases file (action file)**

## Implementation with flask –

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. You first import the `Flask` object from the `flask` package. You then use it to create your Flask application instance with the name `app`. You pass the special variable `__name__` that holds the name of the current Python module. It's used to tell the instance where it's located—you need this because Flask sets up some paths behind the scenes. Once you create the `app` instance, you use it to handle incoming web requests and send responses to the user. `@app.route` is a decorator that turns a regular Python function into a Flask *view function*, which converts the function's return value into an HTTP response to be displayed by an HTTP client, such as a web browser. You pass the value `'/'` to `@app.route()` to signify that this function will respond to web requests for the URL `/`, which is the main URL.

Flask provides a `render_template()` helper function that allows use of the Jinja template engine. This will make managing HTML much easier by writing your HTML code in `.html` files as well as using logic in your HTML code. You'll use these HTML files, (*templates*) to build all of your application pages, such as the main page where you'll

display the current blog posts, the page of the blog post, the page where the user can add a new post, and so on.

## Challenges faced –

The main challenge for us was to make sure that whatever data we were scrapping was accurate so that users don't have to face any issues later. Scraping the data regarding the number of cases took us a little time as the website on which we were taking our data was removed but our team resolved that issue as soon as we could.

## Improvements performed throughout the project –

By 6<sup>th</sup> January, we had our problem statement ready, and also, we planned the course of action to complete this project. We had a basic idea behind the technologies which we will be using.

By 18<sup>th</sup> January, we had our chatbot ready but had to make a few changes to make it more accurate by adding more intents.

By 25<sup>th</sup> January, we have our chatbot up and running. we used flask which is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.

## Future Scope -

In the foreseeable future, our team will try to add more features like sending data to email, data regarding maps and nearby hospitals and make it more interactive.

Code link- [https://github.com/ShubhamSalokhe/Covid19\\_Rasa\\_chatbot.git](https://github.com/ShubhamSalokhe/Covid19_Rasa_chatbot.git)

# Final Result –

## Chat with Covid 19 Chatbot

Other symptoms that are less common and may affect some patients include

1. Loss of taste or smell
2. Nasal congestion
3. Conjunctivitis (also known as red eyes)
4. Sore throat
5. Headache
6. Muscle or joint pain
7. Different types of skin rash
8. Nausea or vomiting
9. Diarrhea
10. Chills or dizziness.

If you have any mild symptoms please isolate yourself and if it is serious please seek medical help.

less than a minute

doses in delhi

less than a minute

State Name : Delhi  
Total First Doses : 1,61,31,029  
Total Second Doses : 1,22,10,958  
First Doses (15-18 year) : 7,74,520  
Precaution Doses : 2,34,766  
Overall Total Vaccination done : 2,93,51,273

SUBMIT





