

Covid-19 Dialog Flow Chatbot Artificial Intelligence

Project Report 2021



Group Members:

- Muhammad Sheroz 9852
- Kehkashan Akram 9825

Class id: 106266

Content:

- 1.0. Introduction to problem**
 - 1.1. Introduction**
 - 1.2. Description**
 - 1.2.1. What is ChatBot?**
 - 1.2.2. What is DialogFlow?**
 - 1.3. Software Requirements**
- 2.0. Project Planing**
 - 2.1. Expected Result**
 - 2.2. Code Language**
 - 2.3. GitHub Link**
 - 2.4. Project Planning**
- 3.0. Progress Table**
- 4.0. Basic ChatBot Flow**
- 5.0. DialogFlow**
 - 5.1. Intent**
 - 5.1.1. Welcome intent**
 - 5.2. Entities**
 - 5.3. Knowledge Base**
 - 5.4. Covid-19 FAQs**
 - 5.5. Fulfilment**
 - 5.6. Integration**
- 6.0. Heroku**
 - 6.1. Deployment Log**
 - 6.2. Build Log**
- 7.0. Rapid API**
- 8.0. MongoDB**
 - 8.1. Cluster**
 - 8.2. Network Access**
 - 8.3. DataBase Access**
 - 8.4. Covid-19 Chat Record Collection**
- 9.0. GitHub**
 - 9.1. Repository**
 - 9.2. Commit Log**
- 10.0. Code**
 - 10.1. Code File directory**
 - 10.2. Requiriments**

- 10.3. Runtime
- 10.4. Procfile
- 10.5. App.py
- 10.6. Make API Request
- 10.7. Result
- 10.8. Template Reaader
- 10.9. Test Mongo
- 10.10. Email
- 10.11. MainFest
- 10.12. Graphs
- 10.13. Conversation
- 11.0. Creating ChatBot
- 12.0. Output of our Project
- 13.0. Conculusion

1.0. Introduction to problem

1.1. Introduction:

In this project we creating chatbot dialog flow. It is artificial intelligent awareness system about Covid-19. One can ask questions about covid-19 and it will answer.

1.2. Description:

We are building a chatbot that will give its users awareness about covid-19.

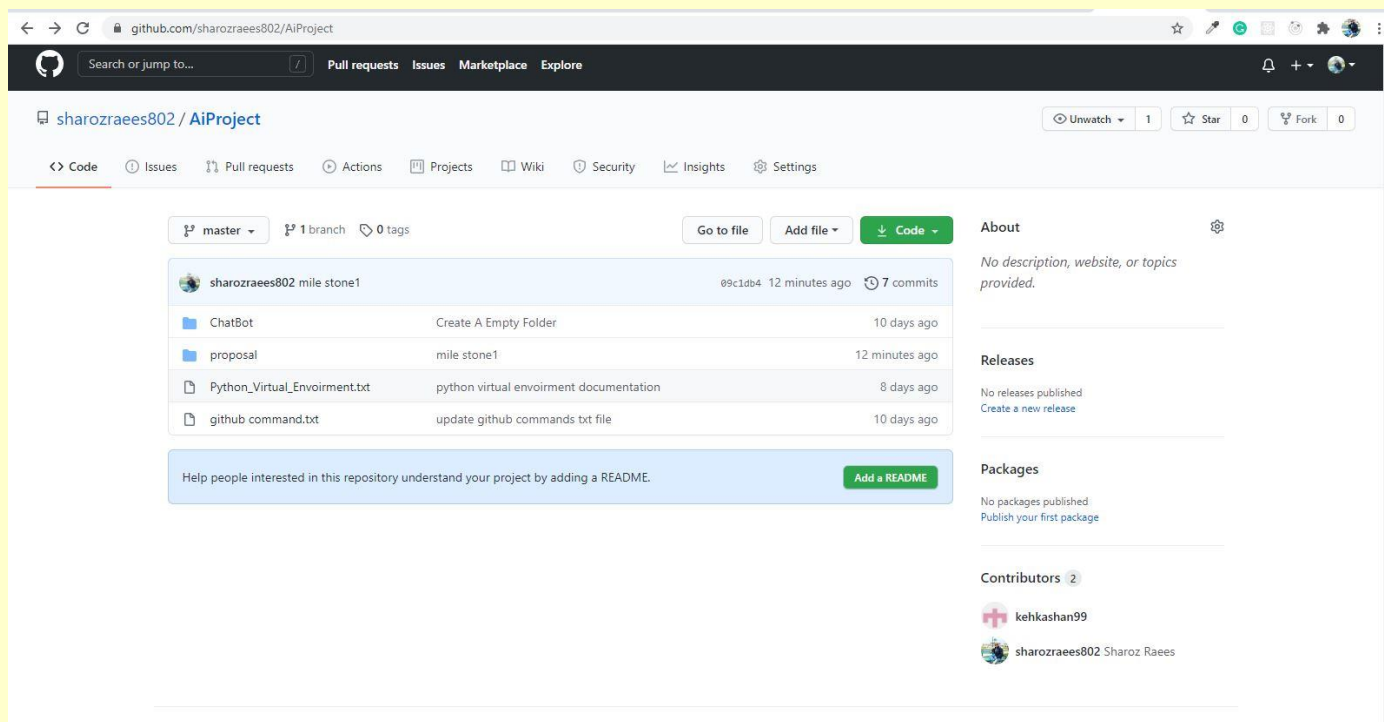
1.2.1. What is a Chatbot?

Interacts through instant messaging, artificially replicating the patterns of human interactions.

1.2.2. What is dialog flow?

Dialog flow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems and related uses.

GitHub Picture

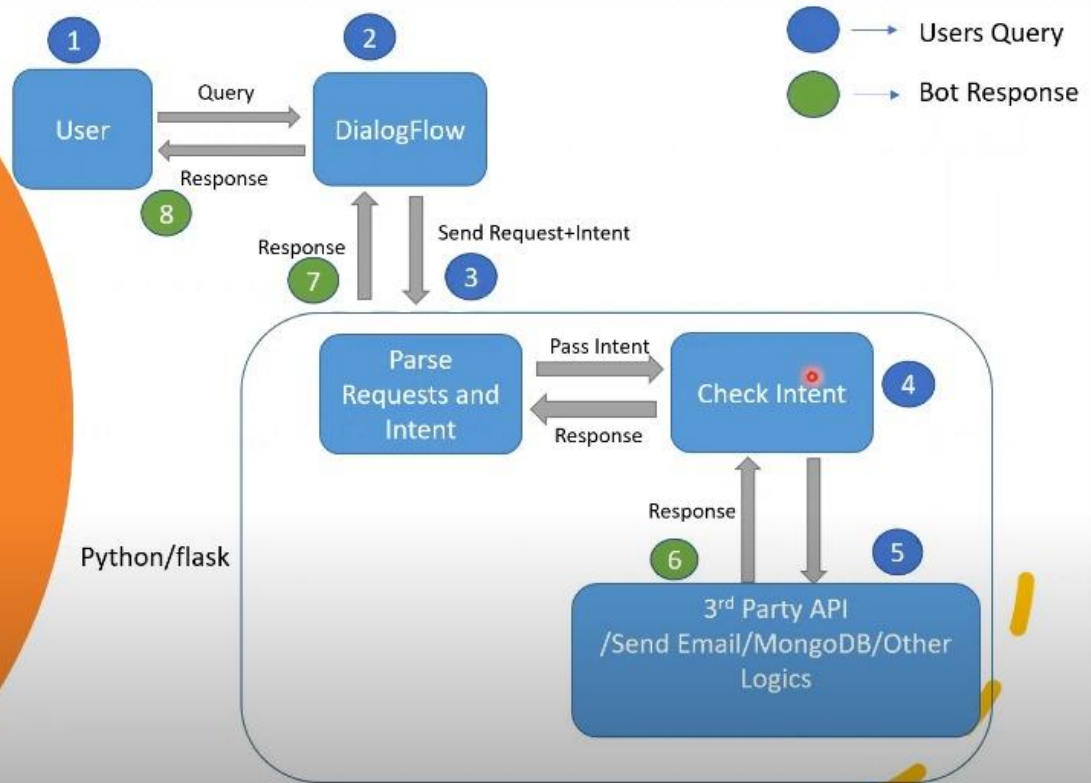


3.0. Progress Table:

	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9
Google, dialogflow, NLU	✓	✓	✓						
Flask/Python, API, Mongo DB				✓	✓	✓			
Testing & Deployment							✓	✓	✓

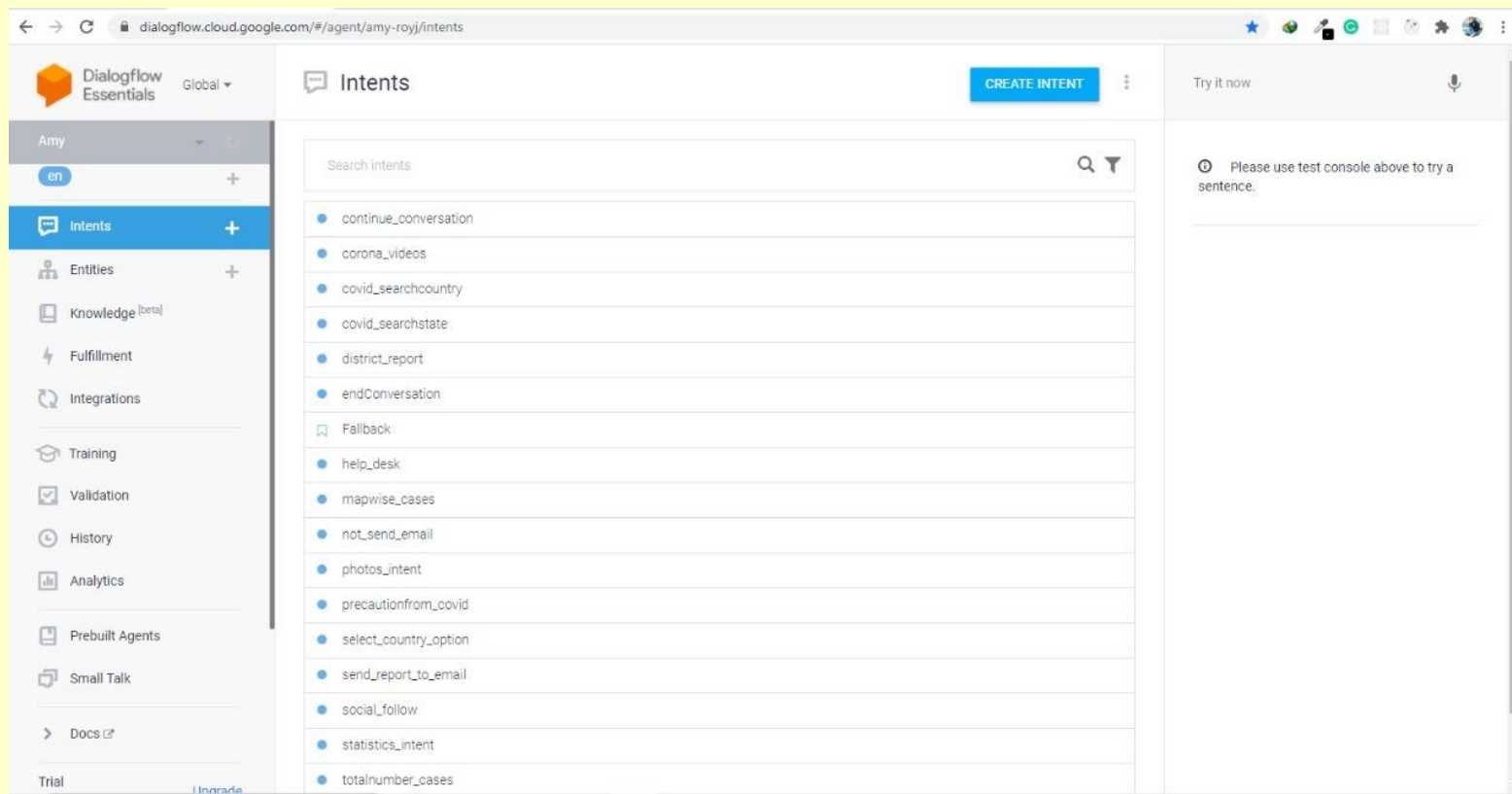
4.0. Basic ChatBot Flow:

Basic ChatBot Flow



5.0. DialogFlow:

5.1. Intents:



The screenshot shows the DialogFlow Intents page in a web browser. The URL is dialogflow.cloud.google.com/#/agent/amy-royj/intents. The page has a sidebar on the left with navigation options: Amy, Intents (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, Docs, and Trial. The main area is titled 'Intents' and contains a search bar and a list of intents. A 'CREATE INTENT' button is in the top right. The right sidebar shows a 'Try it now' section with a microphone icon and a message: 'Please use test console above to try a sentence.'

Search intents
continue_conversation
corona_videos
covid_searchcountry
covid_searchstate
district_report
endConversation
Fallback
help_desk
mapwise_cases
not_send_email
photos_intent
precautionfrom_covid
select_country_option
send_report_to_email
social_follow
statistics_intent
totalnumber_cases

5.1.1. welcome Intent:

Dialogflow Essentials Global

Amy en

Intents +

Entities +

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Small Talk

Docs

Trial Upgrade

Welcome SAVE

Contexts

Events

Training phrases

Search training phrases

Try it now

Please use test console above to try a sentence.

Dialogflow Essentials Global

Amy en

Intents +

Entities +

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Small Talk

Docs

Trial Upgrade

Welcome SAVE

+ New parameter

Responses

DEFAULT TELEGRAM +

Text Response

1 Hello, hope you are staying at home and safe | Welcome, 1. cases Country wise 2. cases Pakistan states 3. cases World wide 4. FAQ 5. Exit

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

Fulfillment

Enable webhook call for this intent

Enable webhook call for slot filling

Try it now

Please use test console above to try a sentence.

5.2. Entities:

The screenshot shows the Dialogflow Entities page for an agent named 'Amy'. The left sidebar contains a navigation menu with options: Amy, Intents, Entities (selected), Knowledge [beta], Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, and Docs. The main content area is titled 'Entities' and has a 'CREATE ENTITY' button. It features two tabs: 'Custom' (active) and 'System'. A search bar labeled 'Search entities' is present. Below it, a list shows a single entity '@ ignore'. The right sidebar has a 'Try it now' button and a text input field with the placeholder 'Please use test console above to try a sentence.'

5.3. Knowledge Base:

The screenshot shows the Dialogflow Knowledge Bases page for the same agent 'Amy'. The left sidebar is identical to the previous screenshot, with 'Knowledge [beta]' now selected. The main content area is titled 'Knowledge Bases' and has a 'CREATE KNOWLEDGE BASE' button. It includes a search bar labeled 'Search knowledge bases'. Below the search bar, a list shows a single knowledge base 'covid'. A section titled 'ADJUST KNOWLEDGE RESULTS PREFERENCE' with the subtitle 'When your query also matches an intent, specify how strongly you prefer knowledge results.' contains a slider. The slider is positioned between 'Weaker' and 'Stronger', with a blue dot indicating the current preference level. The right sidebar is also identical to the previous screenshot.

5.4. Covid 19 FAQs:

The screenshot shows the Dialogflow console interface for a knowledge base named 'covid'. The left sidebar contains navigation options: Amy, Intents, Entities, Knowledge (beta), Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, Docs, and Trial. The main area is titled 'covid' and includes a 'SAVE' button. Below the title is a search bar for documents and a table listing documents. The table has columns for Document Name, Knowledge Type, Mime Type, and Source/Path. A document named 'covid_19FAQ' is listed with a 'View Detail' link. Below the table is a '+ New Document' button. The 'Responses' section shows a 'Text Response' configuration with two entries: 1. \$knowledge.Answer[1] and 2. Enter a text response variant. There is an 'ADD RESPONSES' button and a toggle for 'Set this intent as end of conversation'.

Document Name	Knowledge Type	Mime Type	Source/Path
covid_19FAQ (View Detail)	FAQ	text/csv	File uploaded

+ New Document

Responses ⓘ

DEFAULT TELEGRAM +

Text Response ⓘ

- 1 \$knowledge.Answer[1]
- 2 Enter a text response variant

ADD RESPONSES

☐ Set this intent as end of conversation ⓘ

The screenshot shows the Dialogflow console interface for a document named 'covid_19FAQ'. The left sidebar is the same as the previous screenshot. The main area is titled 'covid_19FAQ' and includes a 'SAVE' button. Below the title is a search bar for question answer entries. Below the search bar is a table listing question answer entries. The table has columns for Question, Answer, and Status. Three entries are listed, each with a 'Question', an 'Answer', and a status of 'ENABLED'.

Question	Answer	Status
Question: What is COVID-19?	Answer: COVID-19 is the infectious disease caused by the most recently discovered corona virus. This new virus and disease were unknown before the outbreak began in Wuhan China in December 2019.	ENABLED
Question: Is this infection going to stay or disappear? Will warmer weather help to contain the virus spread? Will it reappear once colder weather returns?	Answer: This is particularly hard to predict. Some viral diseases (e.g. flu) are largely seasonal and tend to spread more easily in winters rather than in the heat of the summer. We have no idea as of now whether COVID-19 will fall into this category. It could vanish altogether after the summer or perhaps more likely scenario – it could appear again in a second wave. We simply don't know yet.	ENABLED
Question: How long does the virus survive on surfaces?	Answer: It is not certain how long the virus that causes COVID-19 survives on surfaces but it seems to behave like other coronaviruses. Studies suggest that coronaviruses (including the new one)	ENABLED

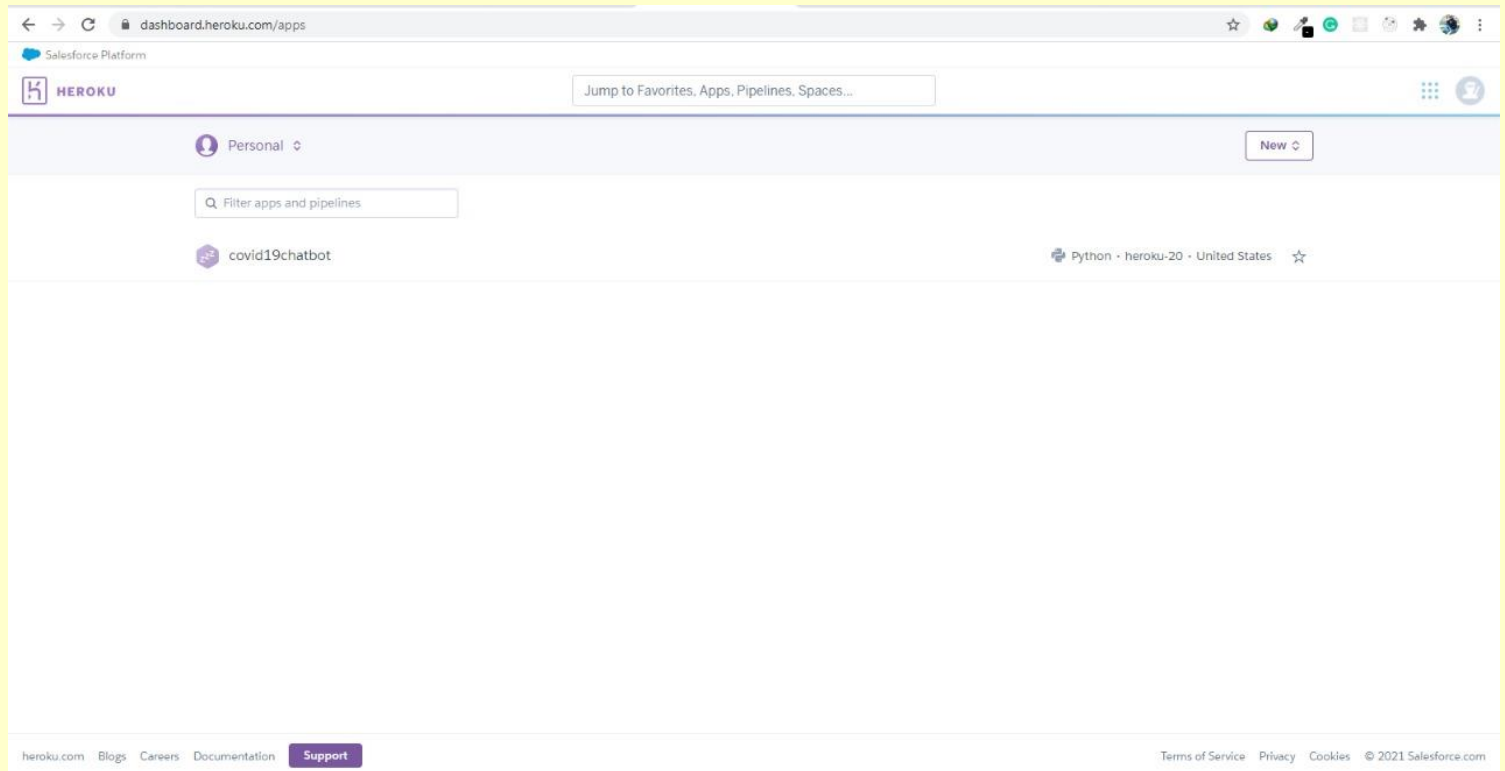
5.5. Fulfilment:

The screenshot shows the Dialogflow Fulfillment configuration page for an agent named 'Amy'. The left sidebar contains navigation links for Intents, Entities, Knowledge, Fulfillment (selected), Integrations, Training, Validation, History, Analytics, Prebuilt Agents, and Small Talk. The main content area is titled 'Fulfillment' and includes a 'Webhook' section with a toggle switch set to 'ENABLED'. Below this, there are fields for 'URL*' (set to 'https://covid19chatbot.herokuapp.com/webhook'), 'BASIC AUTH' (with 'Enter username' and 'Enter password' fields), 'HEADERS' (with 'Enter key' and 'Enter value' fields and an 'Add header' button), and 'SMALL TALK' (with a 'Disable webhook for Smalltalk' dropdown). An 'Inline Editor' section is also present, powered by Google Cloud Functions, with a toggle switch set to 'DISABLED'. A code editor shows the contents of 'index.js' and 'package.json' files. The right sidebar contains a 'Try it now' button and a message: 'Please use test console above to try a sentence.'

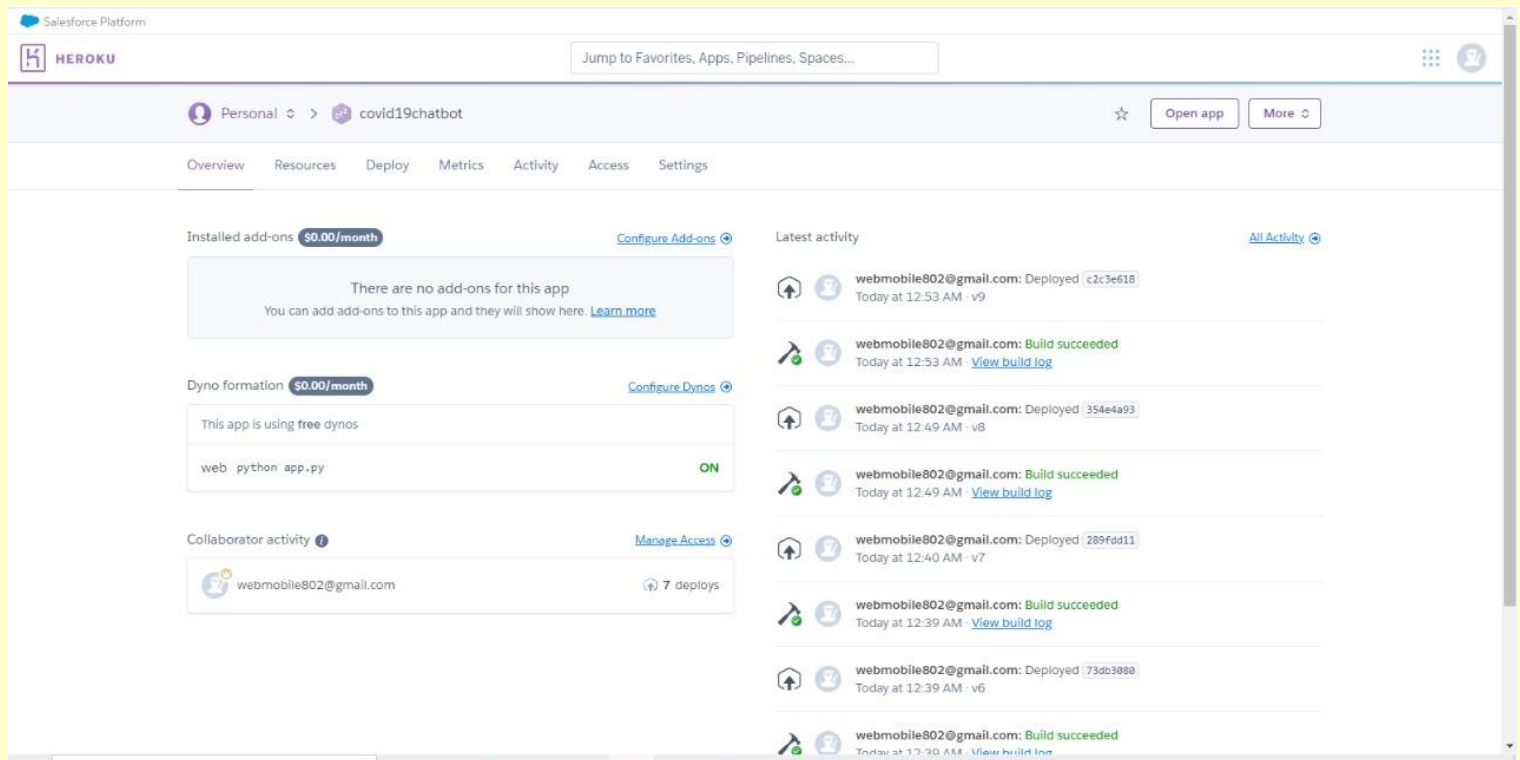
5.6. Integration:

The screenshot shows the Dialogflow Integrations page for an agent named 'Amy'. The left sidebar contains navigation links for Intents, Entities, Knowledge, Fulfillment, Integrations (selected), Training, Validation, History, Analytics, Prebuilt Agents, and Small Talk. The main content area is titled 'Integrations' and is divided into two sections: 'Text based' and 'Open source'. The 'Text based' section contains four integration cards: 'Web Demo', 'Dialogflow Messenger BETA', 'Messenger from Facebook', and 'Workplace from Facebook BETA'. The 'Open source' section contains four integration cards: 'Kik', 'Skype', 'Spark', and 'Twilio IP Messaging'. The right sidebar contains a 'Try it now' button and a message: 'Please use test console above to try a sentence.'

6.0. Heroku:



6.1. Deployment log:



6.2. Build Log:

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal >

covid19chatbot

☆

Open app

More ▾

Overview

Resources

Deploy

Metrics

Activity

Access

Settings

Activity Feed > Build Log

ID 44bf50d5-7701-4bb5-818d-188404034119

```
-----> Building on the Heroku-20 stack
-----> Using buildpack: heroku/python
-----> Python app detected
!     Python has released a security update! Please consider upgrading to python-3.8.9
    Learn More: https://devcenter.heroku.com/articles/python-runtime
-----> No change in requirements detected, installing from cache
-----> Installing pip 20.2.4, setuptools 47.1.1 and wheel 0.36.2
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
      Procfile declares types => web
-----> Compressing...
      Done: 60M
-----> Launching...
      Released v9
      https://covid19chatbot.herokuapp.com/ deployed to Heroku

Build finished
```

heroku.com Blogs Careers Documentation Support

Terms of Service Privacy Cookies © 2021 Salesforce.com

7.0. Rapid API:

The screenshot shows the RapidAPI interface for the COVID-19 API. The top navigation bar includes the RapidAPI logo, a search bar, and links for Categories, Create Organization, API Marketplace, My Apps, Add Your API, and Docs. The main header for the COVID-19 API displays a virus icon, the name 'COVID-19', a 'FREE' badge, a 'Verified' checkmark, and the provider 'api-sports'. It also shows the API was updated a month ago and is in the 'Health, Fitness' category. Metrics for Popularity (9.9/10), Latency (483ms), and Service Level (100%) are displayed. Below the header, there are tabs for Endpoints, About, Tutorials, Discussions, and Pricing. The 'Endpoints' tab is active, showing a search bar and a list of endpoints: GET Countries, GET History, and GET Statistics. The 'GET Countries' endpoint is selected, showing its description 'Get all available countries' and a 'Test Endpoint' button. Below the description, there is a section for 'Personal Account Web Mobile' with a 'Create Organization' button. The 'Header Parameters' section lists 'X-RapidAPI-Key' and 'X-RapidAPI-Host', both marked as 'REQUIRED'. The 'Code Snippets' tab is active, showing a Python request snippet. The snippet includes the URL 'https://covid-193.p.rapidapi.com/countries', headers for 'x-rapidapi-key' and 'x-rapidapi-host', and a request method of 'GET'. The snippet also shows the response handling logic.

Search APIs

Categories

Create Organization

API Marketplace

My Apps

Add Your API

Docs

COVID-19 **FREE** **Verified**

By **api-sports** | Updated a month ago | Health, Fitness

Popularity 9.9/10

Latency 483ms

Service Level 100%

Endpoints About Tutorials Discussions Pricing

Search endpoints

GET Countries

Test Endpoint

Get all available countries

Personal Account Web Mobile

Using this to build something with a team? Organizations make collaboration possible. [Create Organization](#)

RapidAPI App default-application_5134619

REQUIRED

Header Parameters

X-RapidAPI-Key bfacd8b9b6msh67ec4b14fc2440ap1a2d33jsn27e3139fe

ENUM REQUIRED

X-RapidAPI-Host covid-193.p.rapidapi.com

ENUM REQUIRED

Code Snippets Example Responses Results

(Python) Requests [Install SDK](#) [Copy Code](#)

```
import requests

url = "https://covid-193.p.rapidapi.com/countries"

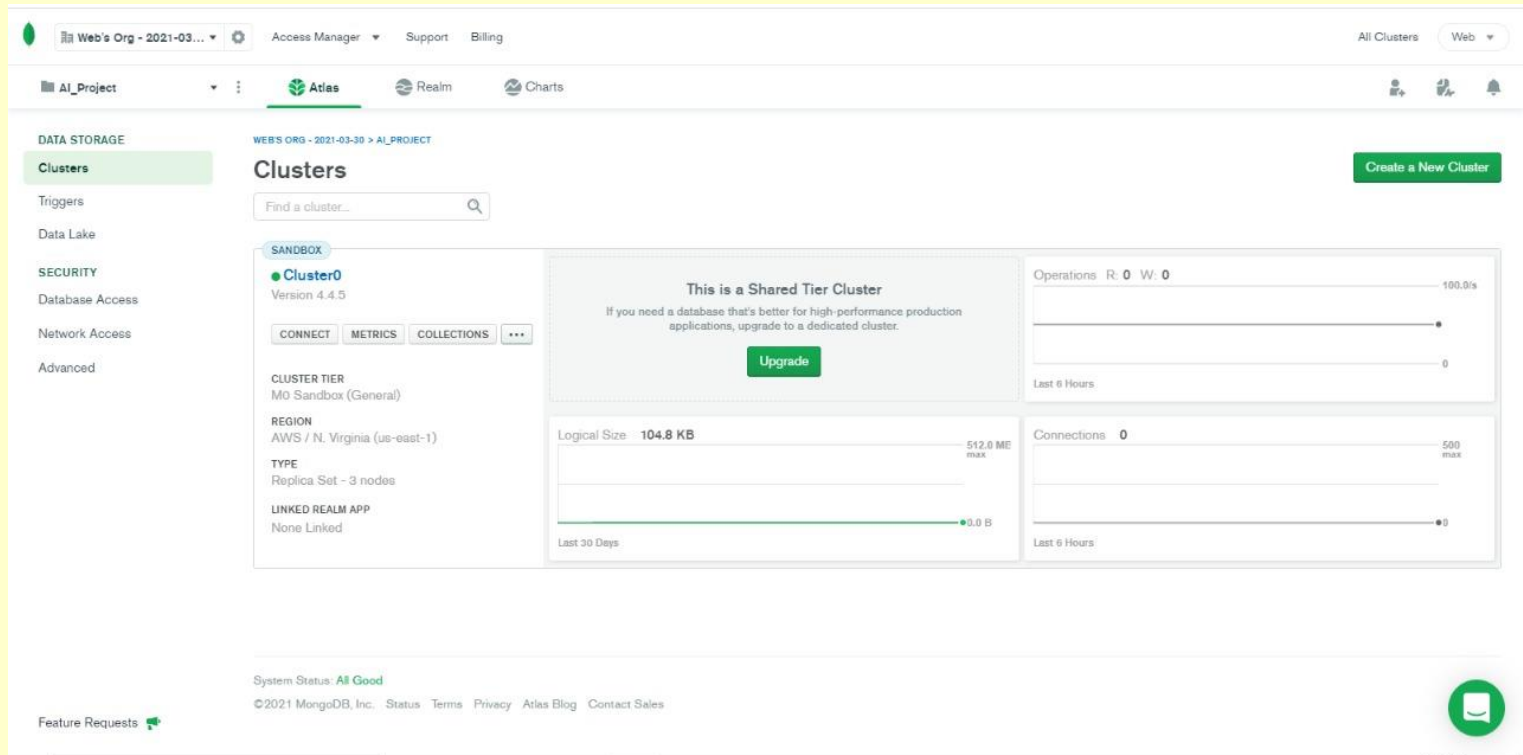
headers = {
    'x-rapidapi-key': "bfacd8b9b6msh67ec4b14fc2440ap1a2d33jsn27e3139fae9e",
    'x-rapidapi-host': "covid-193.p.rapidapi.com"
}

response = requests.request("GET", url, headers=headers)

print(response.text)
```

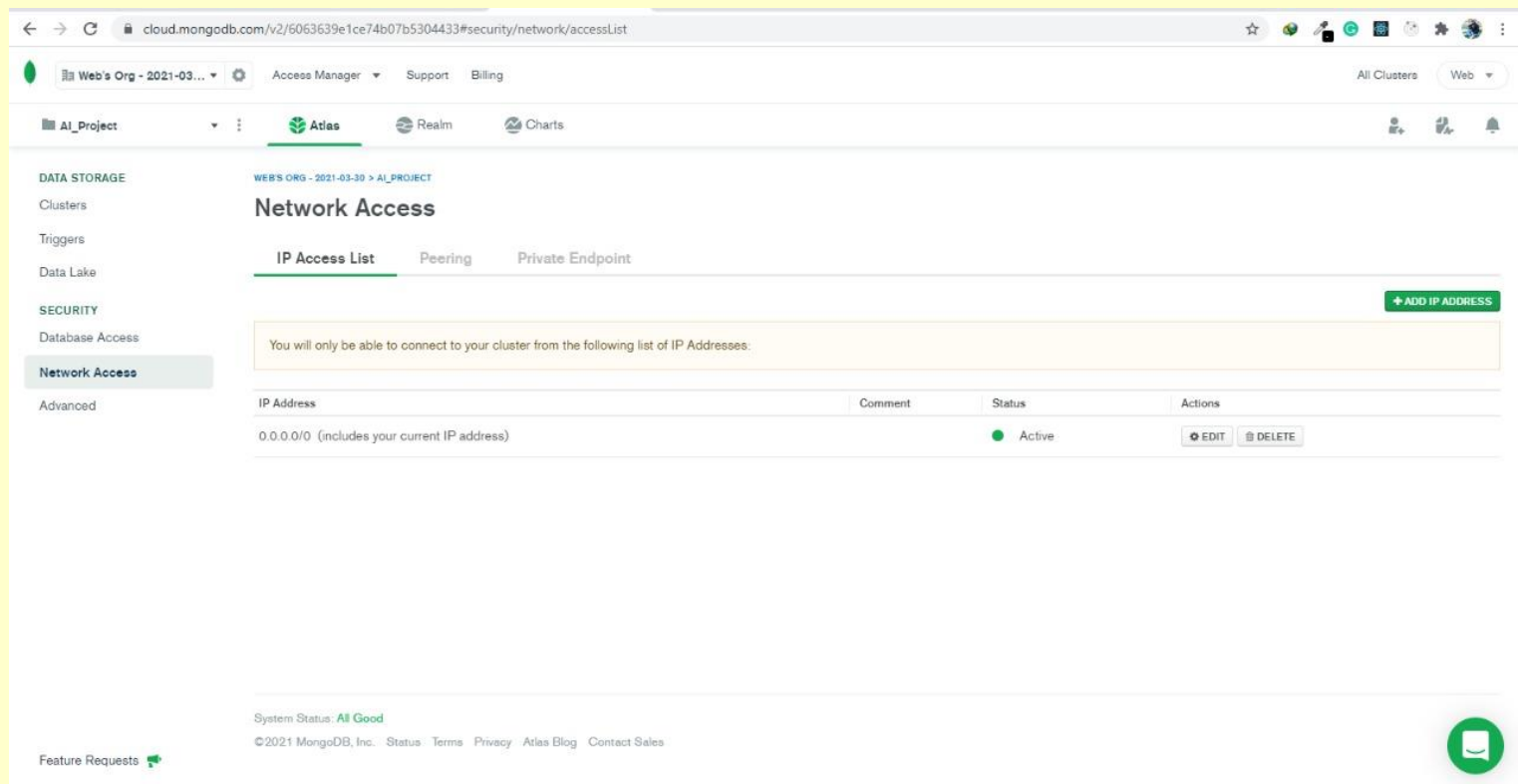
8.0. MongoDB:

8.1. Cluster:



The screenshot shows the MongoDB Atlas interface for a cluster named 'Cluster0'. The left sidebar contains navigation links for DATA STORAGE (Clusters, Triggers, Data Lake), SECURITY (Database Access, Network Access, Advanced), and a Feature Requests link. The main content area is titled 'Clusters' and includes a search bar. Below the search bar, there's a section for 'Cluster0' (Version 4.4.5) with buttons for CONNECT, METRICS, and COLLECTIONS. To the right of this section, a message states 'This is a Shared Tier Cluster' with an 'Upgrade' button. Further right, there are three performance graphs: 'Operations' (R: 0, W: 0, 100.0% max), 'Logical Size' (104.8 KB, 512.0 MB max), and 'Connections' (0, 500 max). All graphs show data for the 'Last 6 Hours'. At the bottom, the system status is 'All Good' and the footer includes copyright information for MongoDB, Inc. (©2021).

8.2. Network Access:



The screenshot shows the MongoDB Atlas 'Network Access' page. The left sidebar is identical to the previous screenshot, with 'Network Access' highlighted under the SECURITY section. The main content area is titled 'Network Access' and has tabs for 'IP Access List', 'Peering', and 'Private Endpoint'. The 'IP Access List' tab is active, showing a message: 'You will only be able to connect to your cluster from the following list of IP Addresses.' Below this message is a table with the following structure:

IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)		Active	EDIT DELETE

At the top right of the table area is a green button labeled '+ ADD IP ADDRESS'. The bottom of the page features the same system status 'All Good' and footer information as the previous screenshot.

8.3. Database Access:

cloud.mongodb.com/v2/6063639e1ce74b07b5304433#security/database/users

Web's Org - 2021-03-30 > AI_PROJECT

Atlas Realm Charts

DATA STORAGE

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

Database Users Custom Roles

+ ADD NEW DATABASE USER

User Name	Authentication Method	MongoDB Roles	Resources	Actions
admin123	SCRAM	readWriteAnyDatabase@admin	All Resources	EDIT DELETE

System Status: All Good

©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Feature Requests

8.4. Covid 19 chat record collection:

cloud.mongodb.com/v2/6063639e1ce74b07b5304433#metrics/replicaSet/60807cc7a0b239379e38e5a4/explorer/covid19DB/chat_records/find

Web's Org - 2021-03-30 > AI_PROJECT

Atlas Realm Charts

Database Access

Network Access

Advanced

+ Create Database

NAMESPACES

covid19DB

chat_records

covid19DB.chat_records

COLLECTION SIZE: 68.84KB TOTAL DOCUMENTS: 247 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER {"filter": "example"}

FIND RESET

QUERY RESULTS 1-20 OF MANY

```
{
  "_id": ObjectId("6064c1200377a102d339ae39"),
  "sessionId": "4eb2494d-8847-46f3-8d15-30b654fa304c-4b8539db",
  "User Intent": "covid_searchcountry",
  "User": "Pakistan",
  "Bot": "New cases :44757",
  "Active cases : 50397",
  "Critical...": "Date: \"2021-03-31/23:40:32\"",
  "Date": "2021-03-31/23:40:32"
}
```

```
{
  "_id": ObjectId("606605a4ff3f3ced083be77"),
  "sessionId": "4eb2494d-8847-46f3-8d15-30b654fa304c-4b8539db",
  "User Intent": "covid_searchcountry",
  "User": "Pakistan",
  "Bot": "New cases :44974",
  "Active cases : 53127",
  "Critical...": "Date: \"2021-04-01/22:40:52\"",
  "Date": "2021-04-01/22:40:52"
}
```

Feature Requests

https://cloud.mongodb.com/v2#/org/6063639e3db86e596238bf0a/

9.0. Git Hub:

9.1. Repository:

This screenshot shows the main page of a GitHub repository named 'sharozraees802 / AiProject'. The interface is in dark mode. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Unwatch, Star, and Fork. A secondary navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area features a file browser showing the repository structure: a 'master' branch with 1 branch and 0 tags, and a list of files and folders including 'ChatBot', 'proposal', 'records', 'Python_Virtual_Envoirement.txt', 'github command.txt', and 'projectset.rar'. Each item shows its commit hash, the commit message, and the time since the last commit. A green 'Add a README' button is visible at the bottom of the file list. On the right side, there are sections for 'About' (no description provided), 'Releases' (no releases published), 'Packages' (no packages published), and 'Contributors' (2 contributors: sharozraees802 and kehkashan99).

9.2. Commit Log:

This screenshot displays the commit log for the 'sharozraees802 / AiProject' repository. The interface is in dark mode. The top navigation bar is identical to the previous screenshot. The main content area shows a list of commits, grouped by date. The commits are as follows:

- Commits on Apr 2, 2021:**
 - send email work (kehkashan99, committed 20 days ago, commit hash 02fc791)
- Commits on Apr 1, 2021:**
 - some txt raw data file delete (sharozraees802, committed 20 days ago, commit hash 0d94c75)
 - same file delete and new two file create for deployment (sharozraees802, committed 20 days ago, commit hash 066f664)
 - rapid api for covid19 data (kehkashan99, committed 20 days ago, commit hash 0f88ee3)
- Commits on Mar 31, 2021:**
 - raw response (sharozraees802, committed 21 days ago, commit hash c5abbac)
 - database connection change (sharozraees802, committed 21 days ago, commit hash 203b3b9)
 - mongodb connection string add (sharozraees802, committed 21 days ago, commit hash 4c88ee3)


data request release



 sharozraees802 committed 22 days ago

 [848191d](#) 

Commits on Mar 30, 2021



app python code

 kehkashan99 committed 22 days ago

 [0b2e3ba](#) 

templete css

 kehkashan99 committed 22 days ago

 [df36e39](#) 

templete is created and all related libraries are installed

 kehkashan99 committed 22 days ago

 [102bc6c](#) 


Commits on Mar 25, 2021

milestone 3 pdf formate

 sharozraees802 committed 27 days ago

 [7301f03](#) 

milstone 3 word format



 kehkashan99 committed 27 days ago

 [9f846c2](#) 

Commits on Mar 20, 2021

same change in jason file

 sharozraees802 committed on Mar 20

 [baa7e3f](#) 

Commits on Mar 18, 2021



Covid Record done all Jason file

 sharozraees802 committed on Mar 18

 [1aa0ed2](#) 

covid records changing done 18 files done

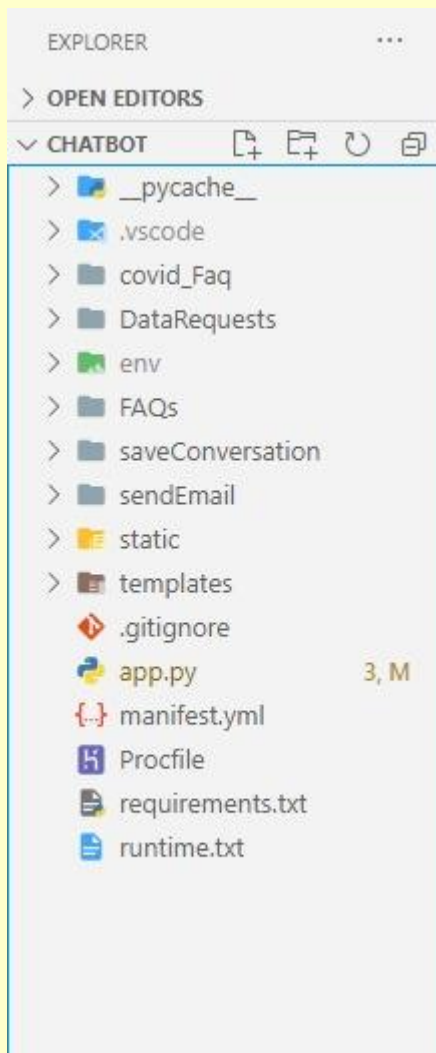
 kehkashan99 committed on Mar 18

 [031745a](#) 

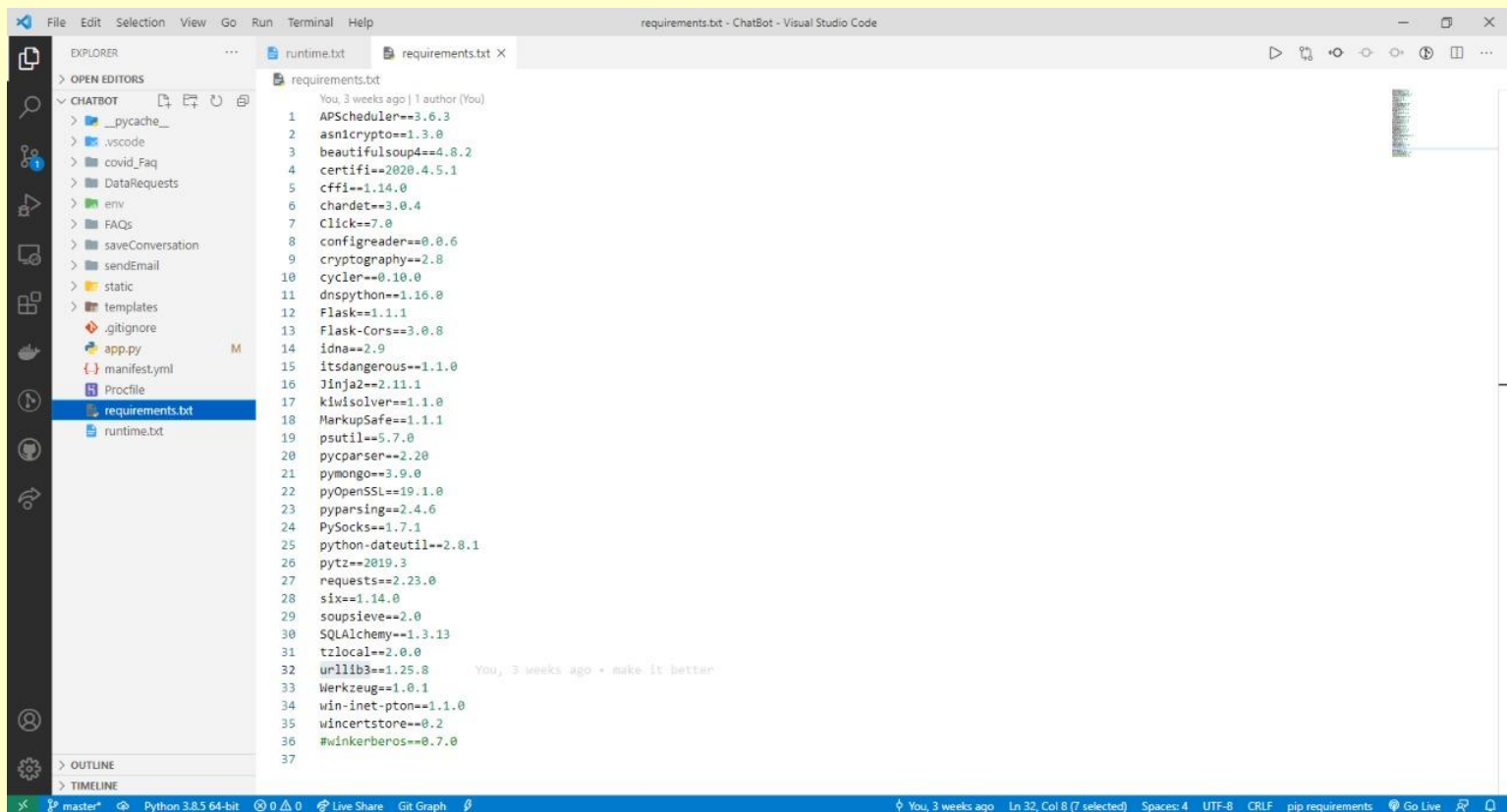
Commits on Mar 17, 2021

10.0. Code:

10.1. Code file directory:



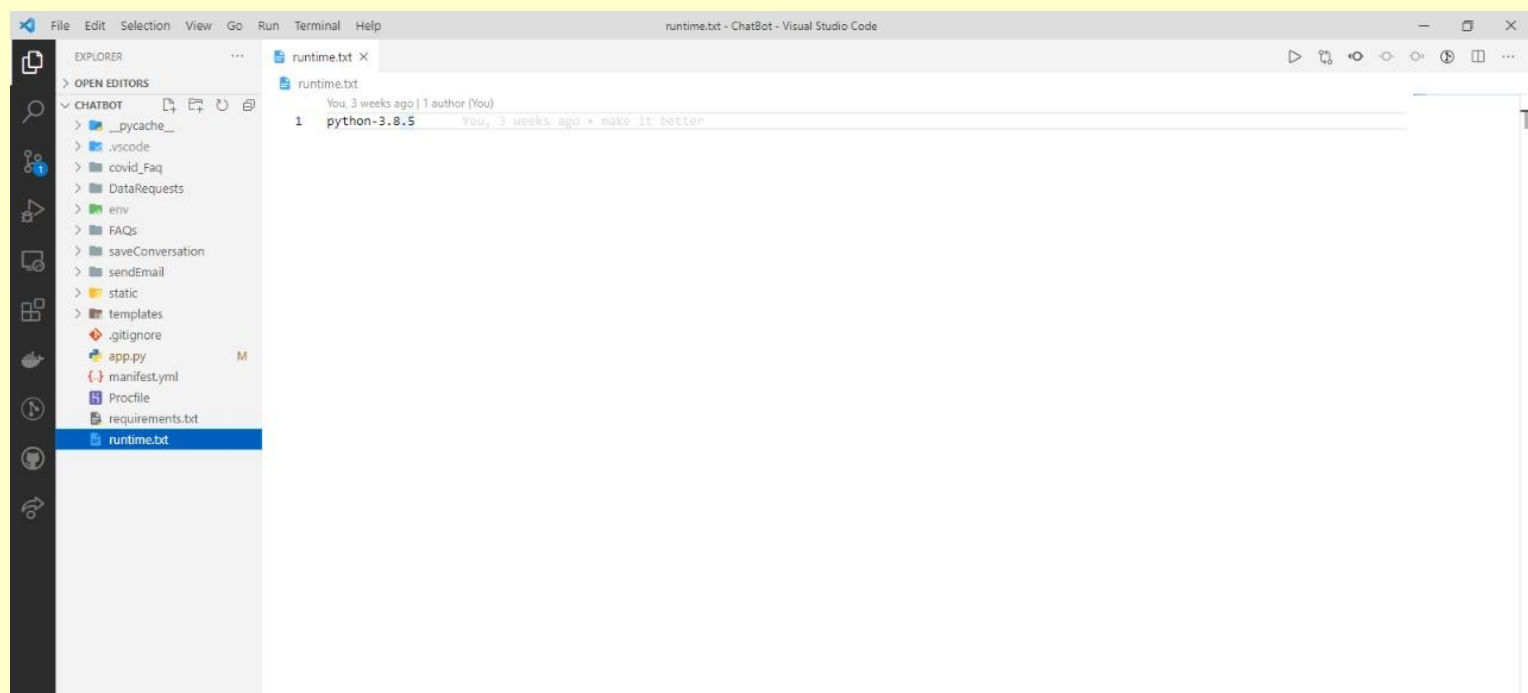
10.2. Requriment:



The screenshot shows the Visual Studio Code interface with the 'requirements.txt' file open in the editor. The file contains a list of Python dependencies. The Explorer sidebar on the left shows the project structure, including folders like 'chatbot', 'covid_Faq', 'DataRequests', 'env', 'FAQs', 'saveConversation', 'sendEmail', 'static', 'templates', and files like '.gitignore', 'app.py', 'manifest.yml', 'Profile', 'requirements.txt', and 'runtime.txt'. The status bar at the bottom indicates the file is 'requirements.txt' and the Python interpreter is 'Python 3.8.5 64-bit'.

```
1 APScheduler==3.6.3
2 asncrypto==1.3.0
3 beautifulsoup4==4.8.2
4 certifi==2020.4.5.1
5 cffi==1.14.0
6 chardet==3.0.4
7 click==7.0
8 configreader==0.0.6
9 cryptography==2.8
10 cyclr==0.10.0
11 dnspython==1.16.0
12 Flask==1.1.1
13 Flask-Cors==3.0.8
14 idna==2.9
15 itsdangerous==1.1.0
16 Jinja2==2.11.1
17 kiwisolver==1.1.0
18 MarkupSafe==1.1.1
19 psutil==5.7.0
20 pycparser==2.20
21 pymongo==3.9.0
22 pyOpenSSL==19.1.0
23 pyparsing==2.4.6
24 PySocks==1.7.1
25 python-dateutil==2.8.1
26 pytz==2019.3
27 requests==2.23.0
28 six==1.14.0
29 soupsieve==2.0
30 SQLAlchemy==1.3.13
31 tzlocal==2.0.0
32 urllib3==1.25.8
33 Werkzeug==1.0.1
34 win-inet-pton==1.1.0
35 wincertstore==0.2
36 #winkerberos==0.7.0
37
```

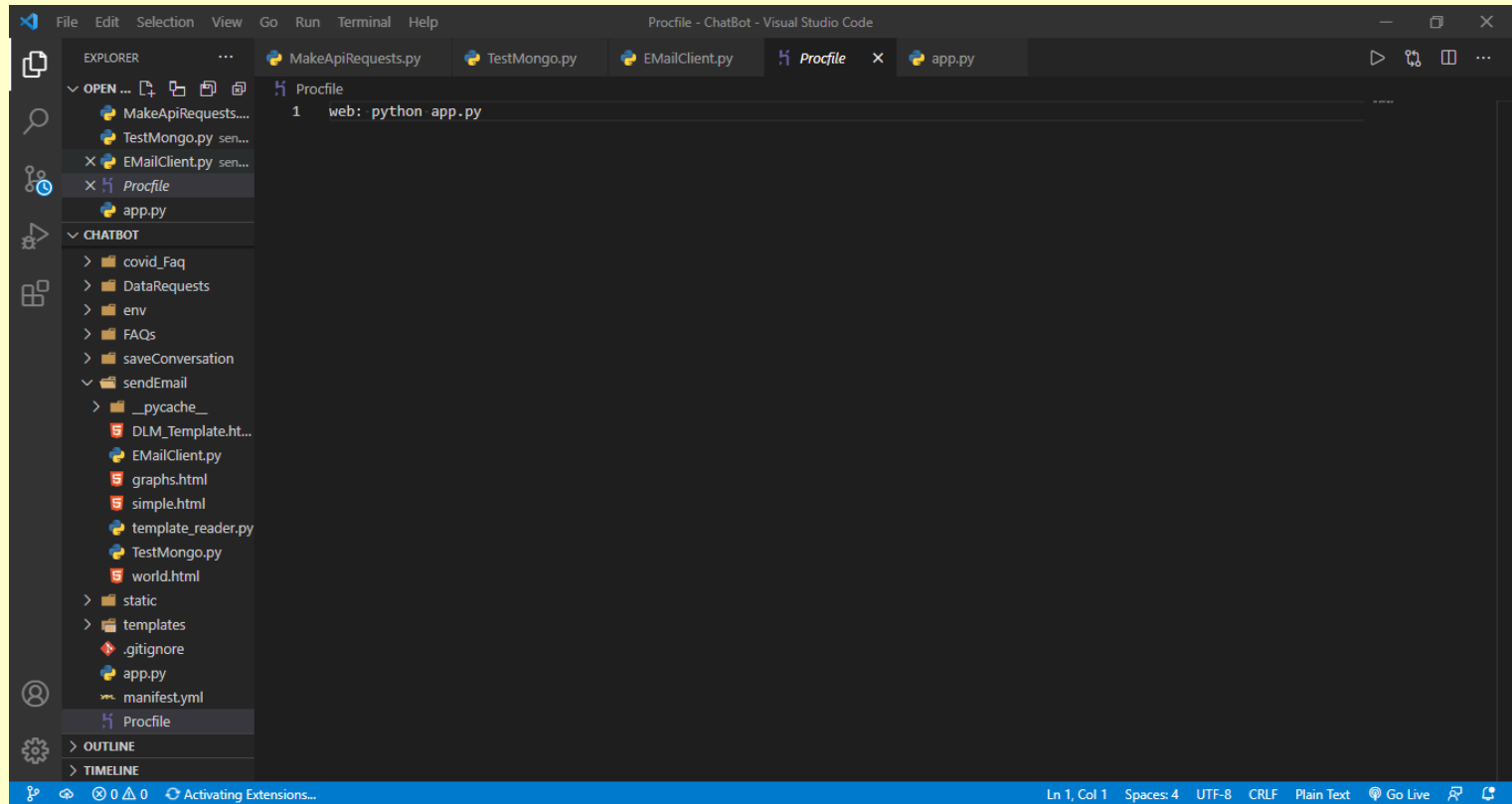
10.3. Runtime:



The screenshot shows the Visual Studio Code interface with the 'runtime.txt' file open in the editor. The file contains a single line of code: 'python-3.8.5'. The Explorer sidebar on the left shows the project structure, including folders like 'chatbot', 'covid_Faq', 'DataRequests', 'env', 'FAQs', 'saveConversation', 'sendEmail', 'static', 'templates', and files like '.gitignore', 'app.py', 'manifest.yml', 'Profile', 'requirements.txt', and 'runtime.txt'. The status bar at the bottom indicates the file is 'runtime.txt' and the Python interpreter is 'Python 3.8.5 64-bit'.

```
1 python-3.8.5
```

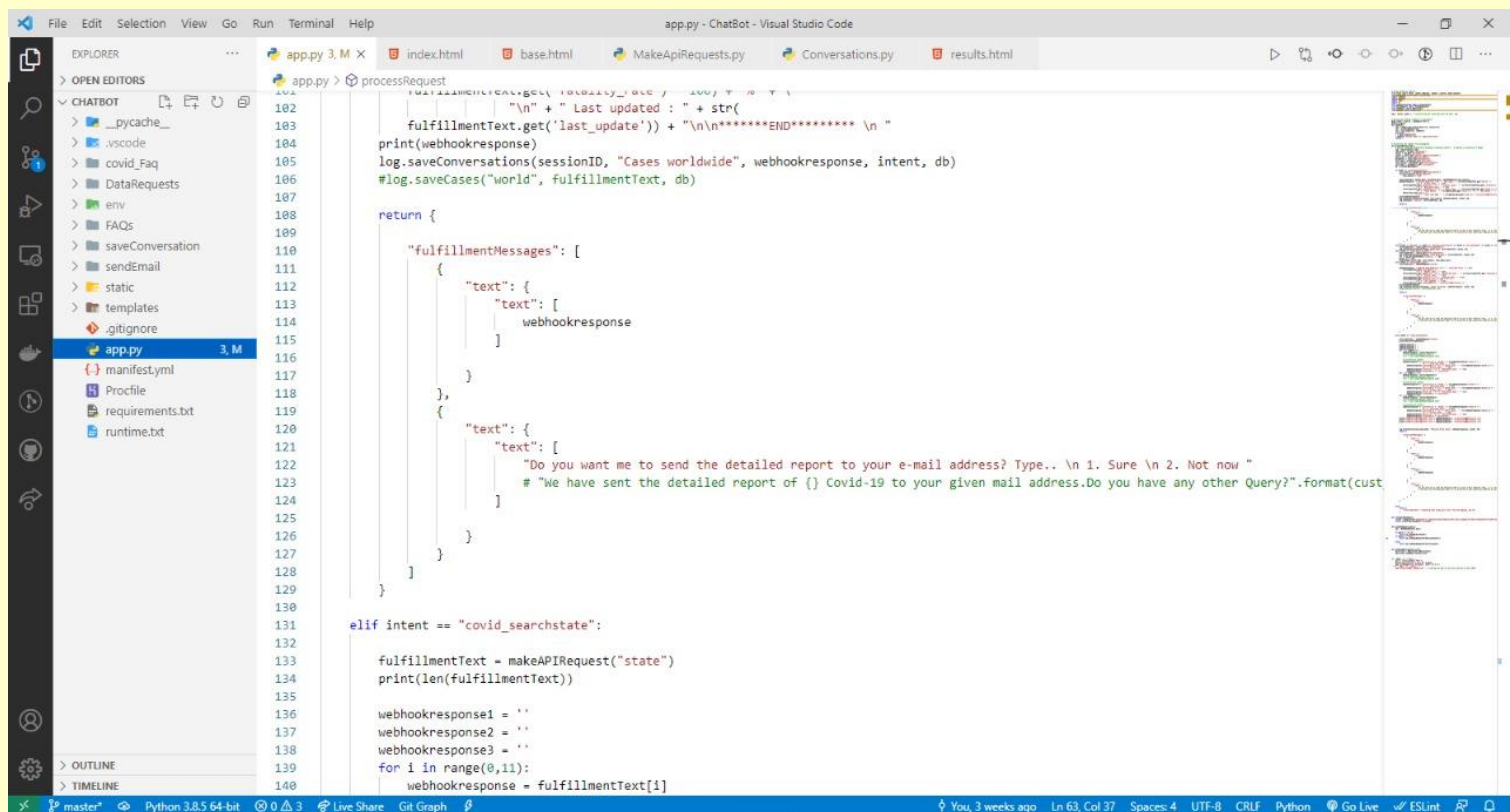
10.4. Procfile:



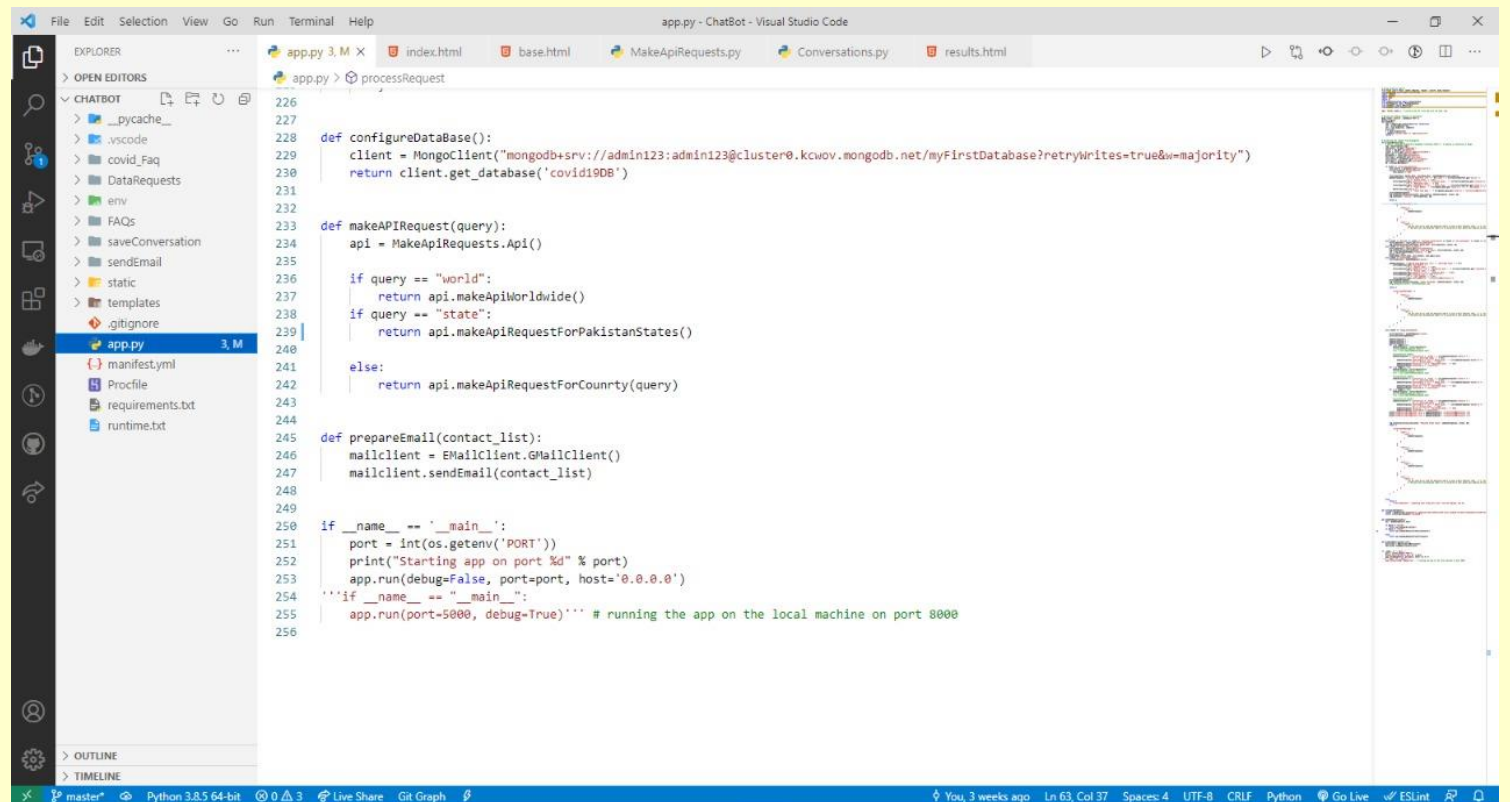
10.5. App.py:

```
1 # doing necessary imports
2 from flask import Flask, render_template, request, jsonify, make_response
3 from flask_cors import CORS, cross_origin
4 import requests
5 import pymongo
6 import json
7 import os
8 from saveConversation import Conversations
9 from DataRequests import MakeApiRequests
10 from sendEmail import EmailClient
11 from pymongo import MongoClient
12
13 app = Flask(__name__) # Initialising the flask app with the name 'app'
14
15 # getting and sending response to dialogflow
16 @app.route('/webhook', methods=['POST'])
17 @cross_origin()
18 def webhook():
19     req = request.get_json(silent=True, force=True)
20     res = processRequest(req)
21     res = json.dumps(res, indent=4)
22     print(res)
23     r = make_response(res)
24     r.headers['Content-Type'] = 'application/json'
25     return r
26
27
28
29 # processing the request from dialogflow
30 def processRequest(req):
31     # dbConn = pymongo.MongoClient("mongodb://localhost:27017/") # opening a connection to Mongo
32     log = Conversations.Log()
33     sessionID = req.get('responseId')
34     result = req.get('queryResult')
35     intent = result.get("intent").get('displayName')
36     query_text = result.get("queryText")
37     parameters = result.get("parameters")
38     cust_name = parameters.get("cust_name")
39     cust_contact = parameters.get("cust_contact")
40     cust_email = parameters.get("cust_email")
41     db = configureDataBase()
42
43     if intent == 'covid_searchcountry':
44         cust_country = parameters.get("geo-country")
45         if(cust_country=="United States"):
46             cust_country = "USA"
47
48         fulfillmentText, deaths_data, testsdone_data = makeAPIRequest(cust_country)
49         webhookresponse = """Covid Report*** \n\n + " New cases : " + str(fulfillmentText.get('new')) + \
50             "\n" + " Active cases : " + str(fulfillmentText.get('active')) + \
51             "\n" + " Recovered cases : " + str(fulfillmentText.get('recovered')) + \
52             "\n" + " Total cases : " + str(fulfillmentText.get('total')) + \
53             "\n" + " Total Deaths : " + str(deaths_data.get('total')) + "\n" + " New Deaths : " + str(
54                 deaths_data.get('new')) + \
55             "\n" + " Total Test Done : " + str(deaths_data.get('total')) + "\n\n*****END***** \n"
56         print(webhookresponse)
57         log.saveConversations(sessionID, cust_country, webhookresponse, intent, db)
58         log.saveCases("country", fulfillmentText, db)
59
60     return {
61         "fulfillmentMessages": [
62             {
63                 "text": fulfillmentText
64             }
65         ]
66     }
```

```
60     return {
61         "fulfillmentMessages": [
62             {
63                 "text": fulfillmentText
64             }
65         ]
66     }
```

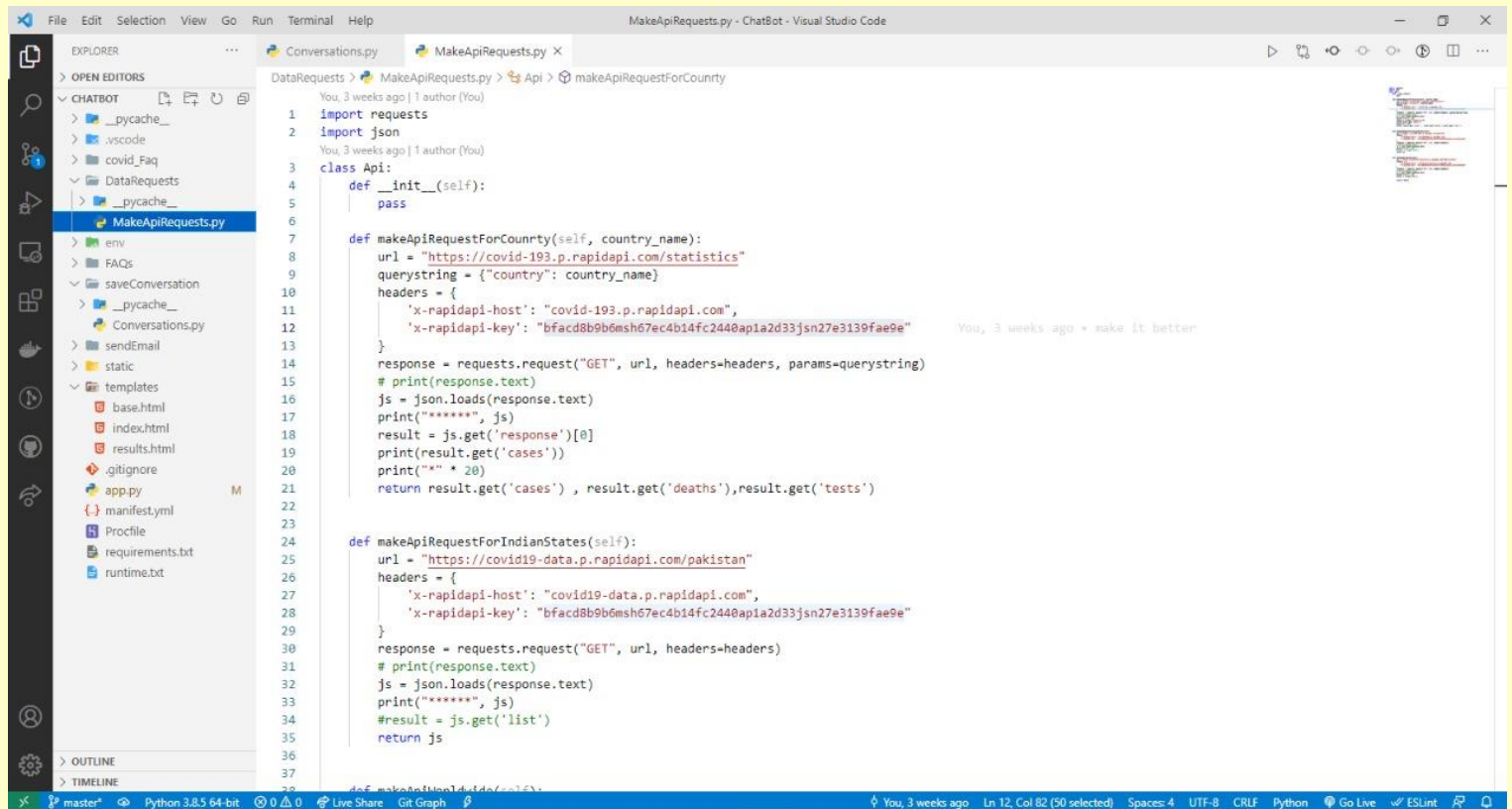



```
102         fulfillmentText.get('last_update')) + "\n" + str(
103         fulfillmentText.get('last_update')) + "\n\n*****END***** \n "
104     print(webhookresponse)
105     log.saveConversations(sessionID, "Cases worldwide", webhookresponse, intent, db)
106     #log.saveCases("world", fulfillmentText, db)
107
108     return {
109
110         "fulfillmentMessages": [
111             {
112                 "text": {
113                     "text": [
114                         webhookresponse
115                     ]
116                 }
117             },
118             {
119                 "text": {
120                     "text": [
121                         "Do you want me to send the detailed report to your e-mail address? Type.. \n 1. Sure \n 2. Not now "
122                         # "We have sent the detailed report of {} Covid-19 to your given mail address.Do you have any other Query?".format(cust
123                     ]
124                 }
125             }
126         ]
127     }
128
129 }
130
131 elif intent == "covid_searchstate":
132
133     fulfillmentText = makeAPIRequest("state")
134     print(len(fulfillmentText))
135
136     webhookresponse1 = ''
137     webhookresponse2 = ''
138     webhookresponse3 = ''
139     for i in range(0,11):
140         webhookresponse = fulfillmentText[i]
```



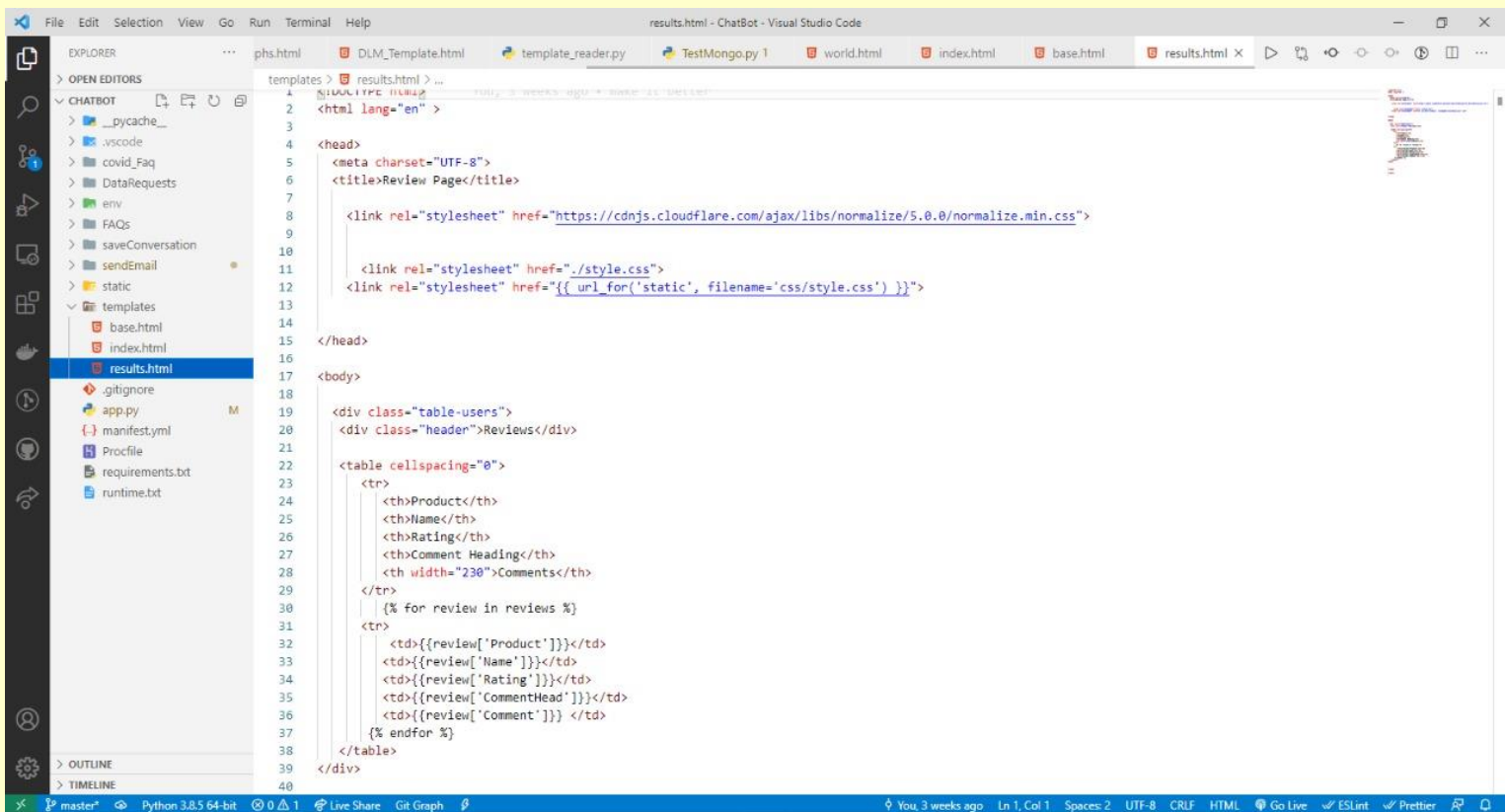
```
226
227
228 def configureDataBase():
229     client = MongoClient("mongodb+srv://admin123:admin123@cluster0.kcwoov.mongodb.net/myFirstDatabase?retryWrites=true&w=majority")
230     return client.get_database('covid19DB')
231
232
233 def makeAPIRequest(query):
234     api = MakeAPIRequests.Api()
235
236     if query == "world":
237         return api.makeApiWorldwide()
238     if query == "state":
239         return api.makeApiRequestForPakistanStates()
240
241     else:
242         return api.makeApiRequestForCountry(query)
243
244
245 def prepareEmail(contact_list):
246     mailclient = EmailClient.EmailClient()
247     mailclient.sendEmail(contact_list)
248
249
250 if __name__ == '__main__':
251     port = int(os.getenv('PORT'))
252     print("Starting app on port %d" % port)
253     app.run(debug=False, port=port, host='0.0.0.0')
254
255 '''if __name__ == "__main__":
256     app.run(port=5000, debug=True)''' # running the app on the local machine on port 8000
```

10.6. Make API Request:



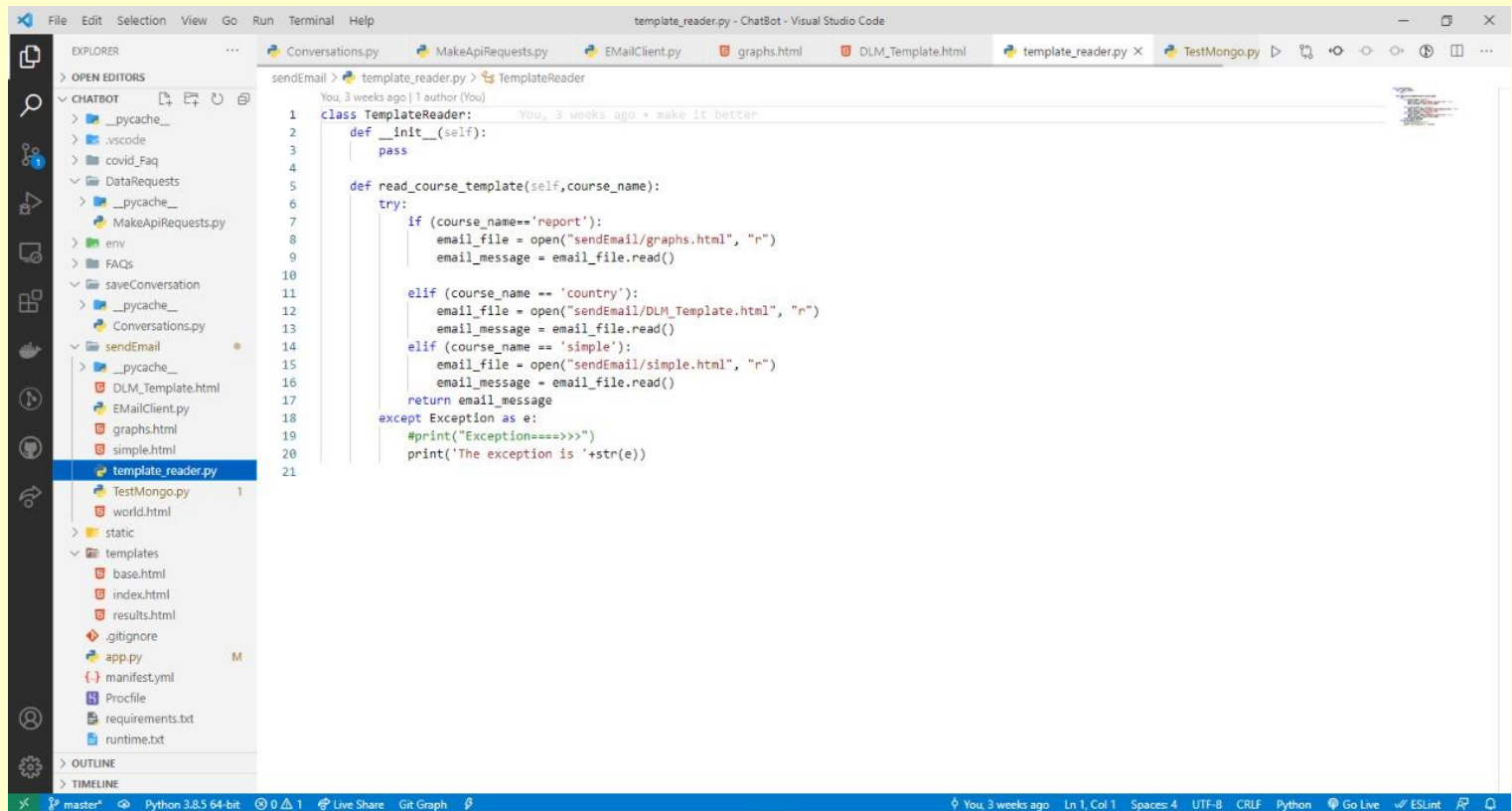
```
1 import requests
2 import json
3
4 class Api:
5     def __init__(self):
6         pass
7
8     def makeApiRequestForCountry(self, country_name):
9         url = "https://covid-193.p.rapidapi.com/statistics"
10        querystring = {"country": country_name}
11        headers = {
12            'x-rapidapi-host': "covid-193.p.rapidapi.com",
13            'x-rapidapi-key': "bfacd8b9b6msh67ec4b14fc2440ap1a2d33jsn27e3139fae9e"
14        }
15        response = requests.request("GET", url, headers=headers, params=querystring)
16        # print(response.text)
17        js = json.loads(response.text)
18        print("*****", js)
19        result = js.get('response')[0]
20        print(result.get('cases'))
21        print("x" * 20)
22        return result.get('cases'), result.get('deaths'), result.get('tests')
23
24    def makeApiRequestForIndianStates(self):
25        url = "https://covid19-data.p.rapidapi.com/pakistan"
26        headers = {
27            'x-rapidapi-host': "covid19-data.p.rapidapi.com",
28            'x-rapidapi-key': "bfacd8b9b6msh67ec4b14fc2440ap1a2d33jsn27e3139fae9e"
29        }
30        response = requests.request("GET", url, headers=headers)
31        # print(response.text)
32        js = json.loads(response.text)
33        print("*****", js)
34        # result = js.get('list')
35        return js
```

10.7. Result:



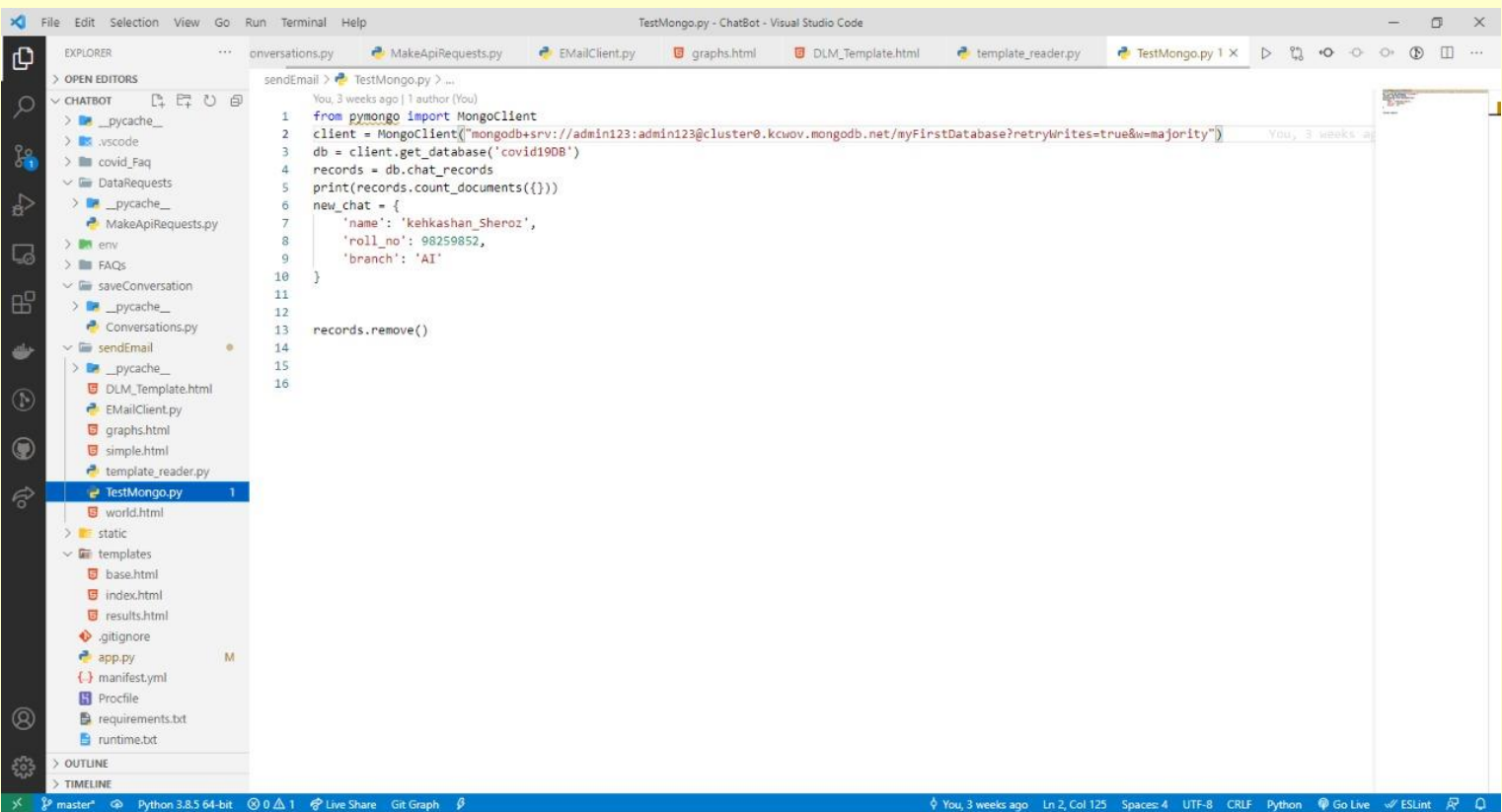
```
1 <html lang="en" >
2
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Review Page</title>
7
8     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
9
10    <link rel="stylesheet" href="./style.css">
11    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
12
13 </head>
14
15 <body>
16
17 <div class="table-users">
18 <div class="header">Reviews</div>
19
20 <table cellpadding="0">
21 <tr>
22 <th>Product</th>
23 <th>Name</th>
24 <th>Rating</th>
25 <th>Comment Heading</th>
26 <th width="230">Comments</th>
27 </tr>
28 <tr>
29 <td>{{ review['Product'] }}</td>
30 <td>{{ review['Name'] }}</td>
31 <td>{{ review['Rating'] }}</td>
32 <td>{{ review['CommentHead'] }}</td>
33 <td>{{ review['Comment'] }}</td>
34 </tr>
35 </table>
36 </div>
```


10.8. Template Reader:



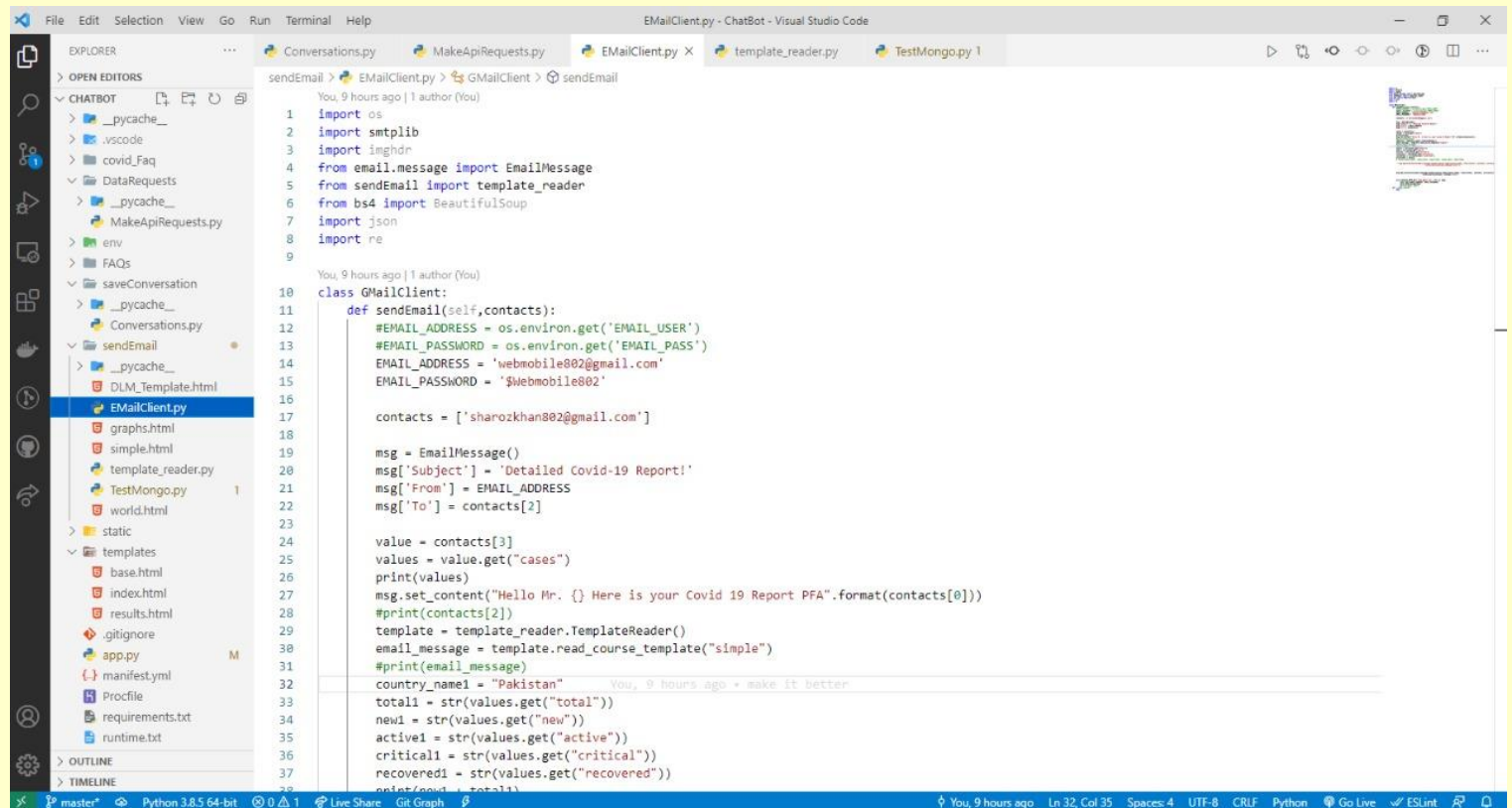
```
1 class TemplateReader:
2     def __init__(self):
3         pass
4
5     def read_course_template(self, course_name):
6         try:
7             if (course_name == 'report'):
8                 email_file = open("sendEmail/graphs.html", "r")
9                 email_message = email_file.read()
10
11             elif (course_name == 'country'):
12                 email_file = open("sendEmail/DLM_Template.html", "r")
13                 email_message = email_file.read()
14             elif (course_name == 'simple'):
15                 email_file = open("sendEmail/simple.html", "r")
16                 email_message = email_file.read()
17             return email_message
18         except Exception as e:
19             #print("Exceptions====>>>")
20             print('The exception is '+str(e))
21
```

10.9. Test Mongo:



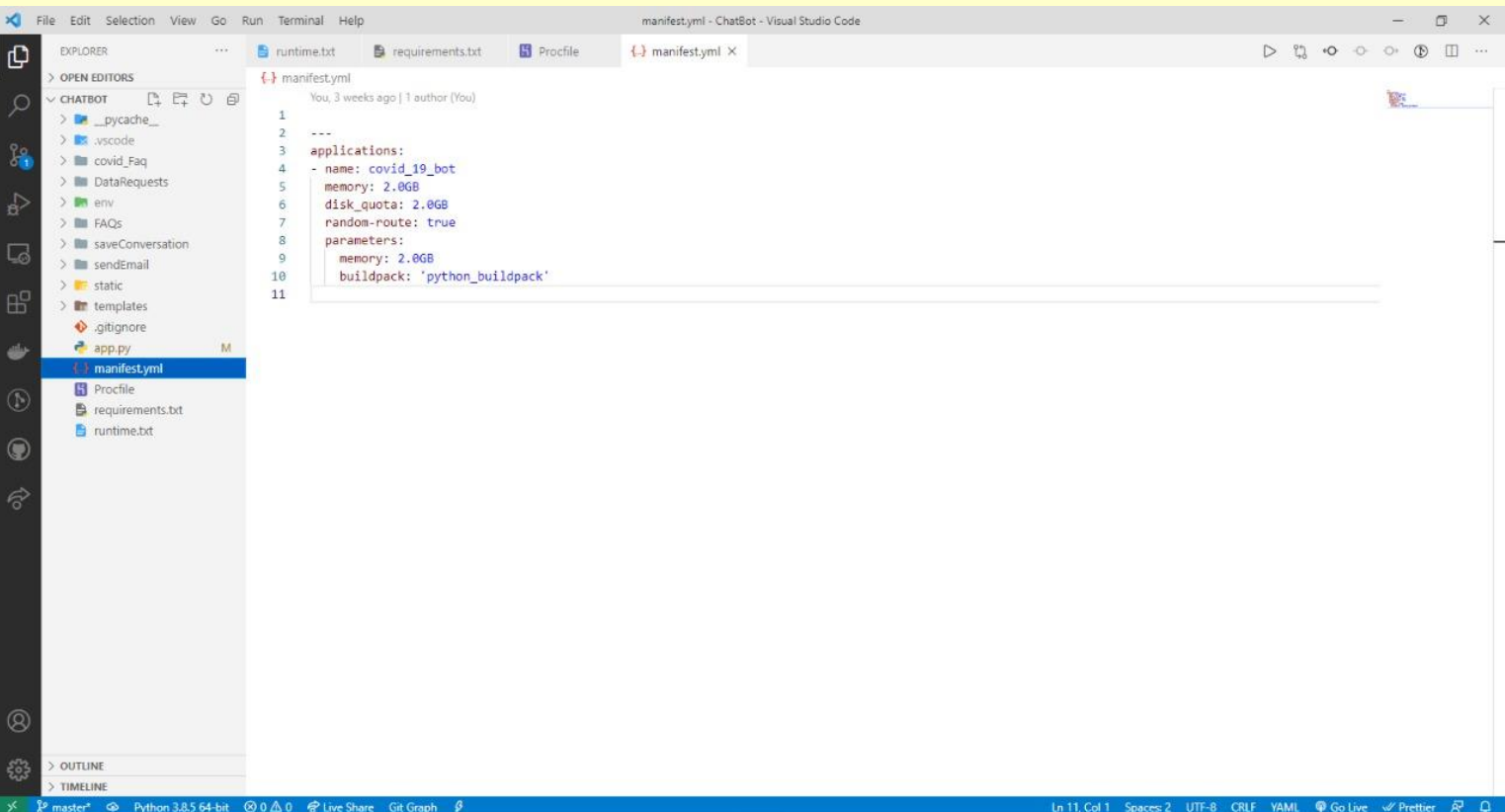
```
1 from pymongo import MongoClient
2 client = MongoClient("mongodb+srv://admin123:admin123@cluster0.kcvov.mongodb.net/myFirstDatabase?retryWrites=true&w=majority")
3 db = client.get_database('covid19DB')
4 records = db.chat_records
5 print(records.count_documents({}))
6 new_chat = {
7     'name': 'kehkashan_Sheroz',
8     'roll_no': 98259852,
9     'branch': 'AI'
10 }
11
12
13 records.remove()
14
15
16
```

10.10. Email:



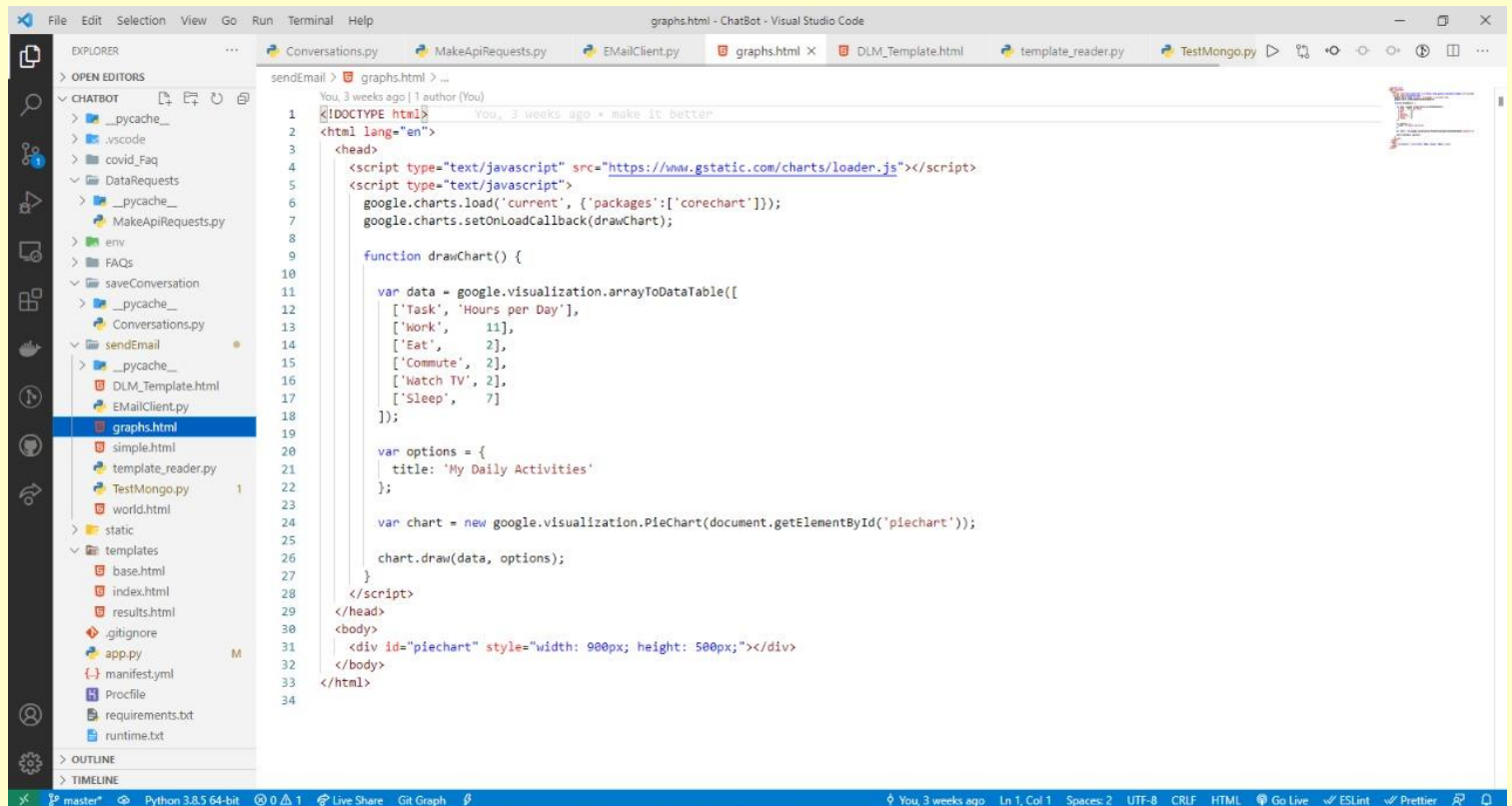
```
1 import os
2 import smtplib
3 import imgndr
4 from email.message import EmailMessage
5 from sendEmail import template_reader
6 from bs4 import BeautifulSoup
7 import json
8 import re
9
10 You, 9 hours ago | 1 author (You)
11
12 class GMailClient:
13     def sendEmail(self, contacts):
14         #EMAIL_ADDRESS = os.environ.get('EMAIL_USER')
15         #EMAIL_PASSWORD = os.environ.get('EMAIL_PASS')
16         EMAIL_ADDRESS = 'webmobile802@gmail.com'
17         EMAIL_PASSWORD = '$Webmobile802'
18
19         contacts = ['sharozkhan802@gmail.com']
20
21         msg = EmailMessage()
22         msg['Subject'] = 'Detailed Covid-19 Report!'
23         msg['From'] = EMAIL_ADDRESS
24         msg['To'] = contacts[2]
25
26         value = contacts[3]
27         values = value.get("cases")
28         print(values)
29         msg.set_content("Hello Mr. {} Here is your Covid 19 Report PFA".format(contacts[0]))
30         #print(contacts[2])
31         template = template_reader.TemplateReader()
32         email_message = template.read_course_template("simple")
33         #print(email_message)
34         country_name1 = "Pakistan"
35         total1 = str(values.get("total"))
36         new1 = str(values.get("new"))
37         active1 = str(values.get("active"))
38         critical1 = str(values.get("critical"))
39         recovered1 = str(values.get("recovered"))
40         print(country_name1, total1, new1, active1, critical1, recovered1)
```

10.11. Manifest:



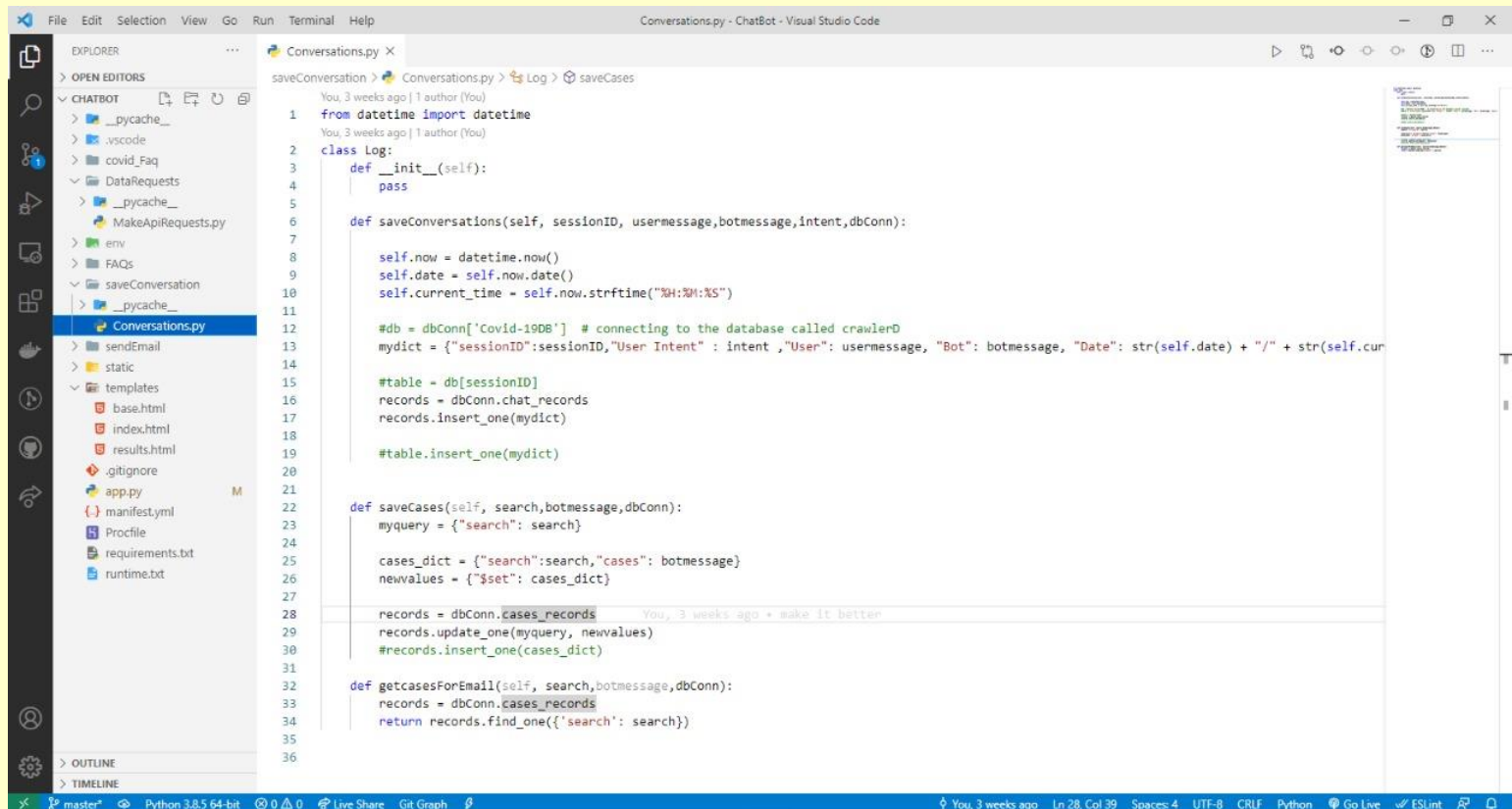
```
1 You, 3 weeks ago | 1 author (You)
2 ---
3 applications:
4   - name: covid_19_bot
5     memory: 2.0GB
6     disk_quota: 2.0GB
7     random-route: true
8     parameters:
9       memory: 2.0GB
10      buildpack: 'python_buildpack'
```

10.12. Graphs:



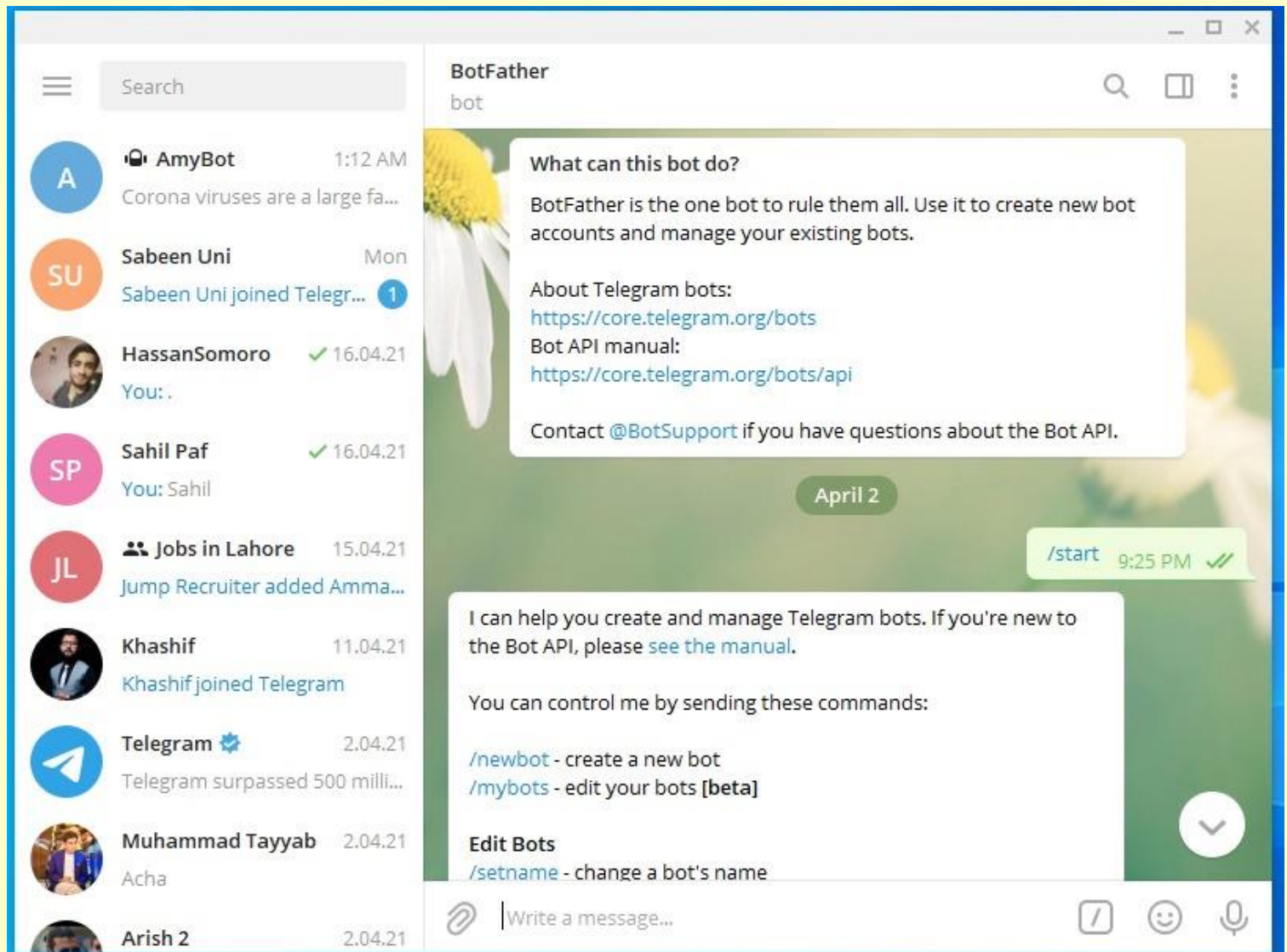
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
5   <script type="text/javascript">
6     google.charts.load('current', {'packages':['corechart']});
7     google.charts.setOnLoadCallback(drawChart);
8
9     function drawChart() {
10
11       var data = google.visualization.arrayToDataTable([
12         ['Task', 'Hours per Day'],
13         ['Work', 11],
14         ['Eat', 2],
15         ['Commute', 2],
16         ['Watch TV', 2],
17         ['Sleep', 7]
18       ]);
19
20       var options = {
21         title: 'My Daily Activities'
22       };
23
24       var chart = new google.visualization.PieChart(document.getElementById('piechart'));
25
26       chart.draw(data, options);
27     }
28   </script>
29 </head>
30 <body>
31   <div id="piechart" style="width: 900px; height: 500px;"></div>
32 </body>
33 </html>
```

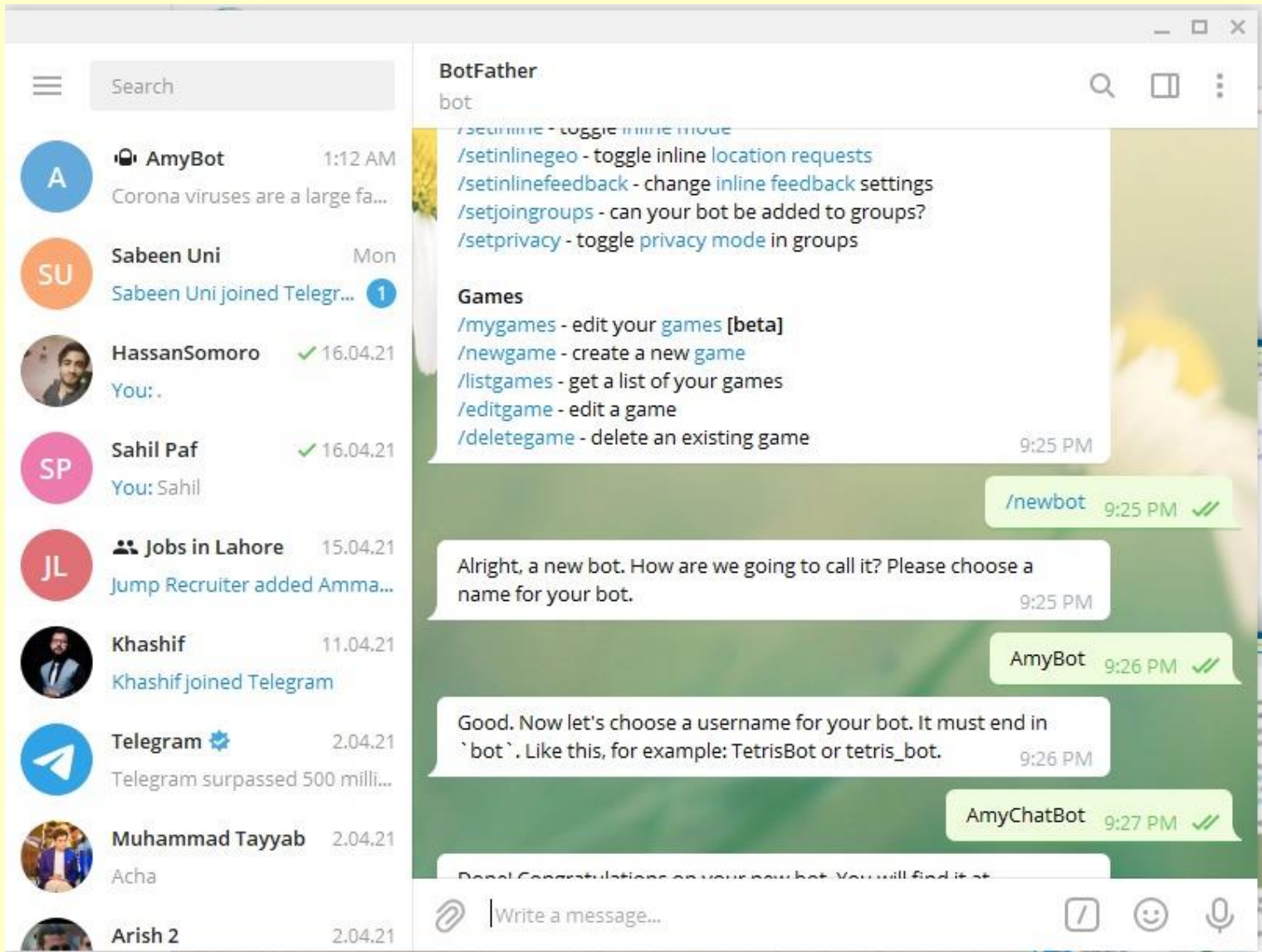
10.13. Conversation:

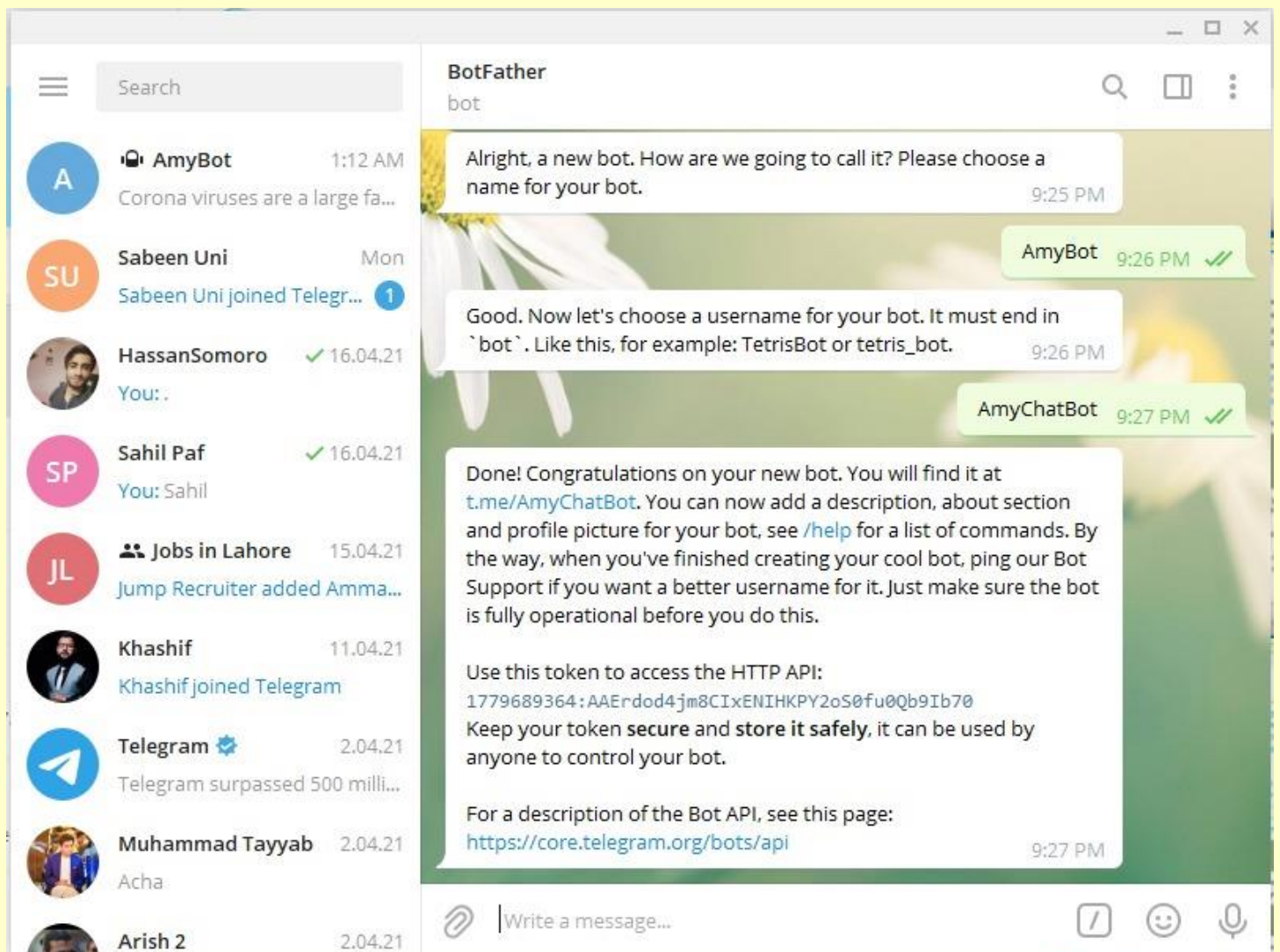


```
1 from datetime import datetime
2
3 class Log:
4     def __init__(self):
5         pass
6
7     def saveConversations(self, sessionId, usermessage, botmessage, intent, dbConn):
8
9         self.now = datetime.now()
10        self.date = self.now.date()
11        self.current_time = self.now.strftime("%H:%M:%S")
12
13        #db = dbConn['Covid-19DB'] # connecting to the database called crawlerDB
14        mydict = {"sessionId":sessionId, "User Intent" : intent, "User": usermessage, "Bot": botmessage, "Date": str(self.date) + "/" + str(self.current_time)}
15
16        #table = db[sessionId]
17        records = dbConn.chat_records
18        records.insert_one(mydict)
19
20        #table.insert_one(mydict)
21
22    def saveCases(self, search, botmessage, dbConn):
23        myquery = {"search": search}
24
25        cases_dict = {"search":search, "cases": botmessage}
26        newvalues = {"$set": cases_dict}
27
28        records = dbConn.cases_records
29        records.update_one(myquery, newvalues)
30        #records.insert_one(cases_dict)
31
32    def getCasesForEmail(self, search, botmessage, dbConn):
33        records = dbConn.cases_records
34        return records.find_one({'search': search})
```

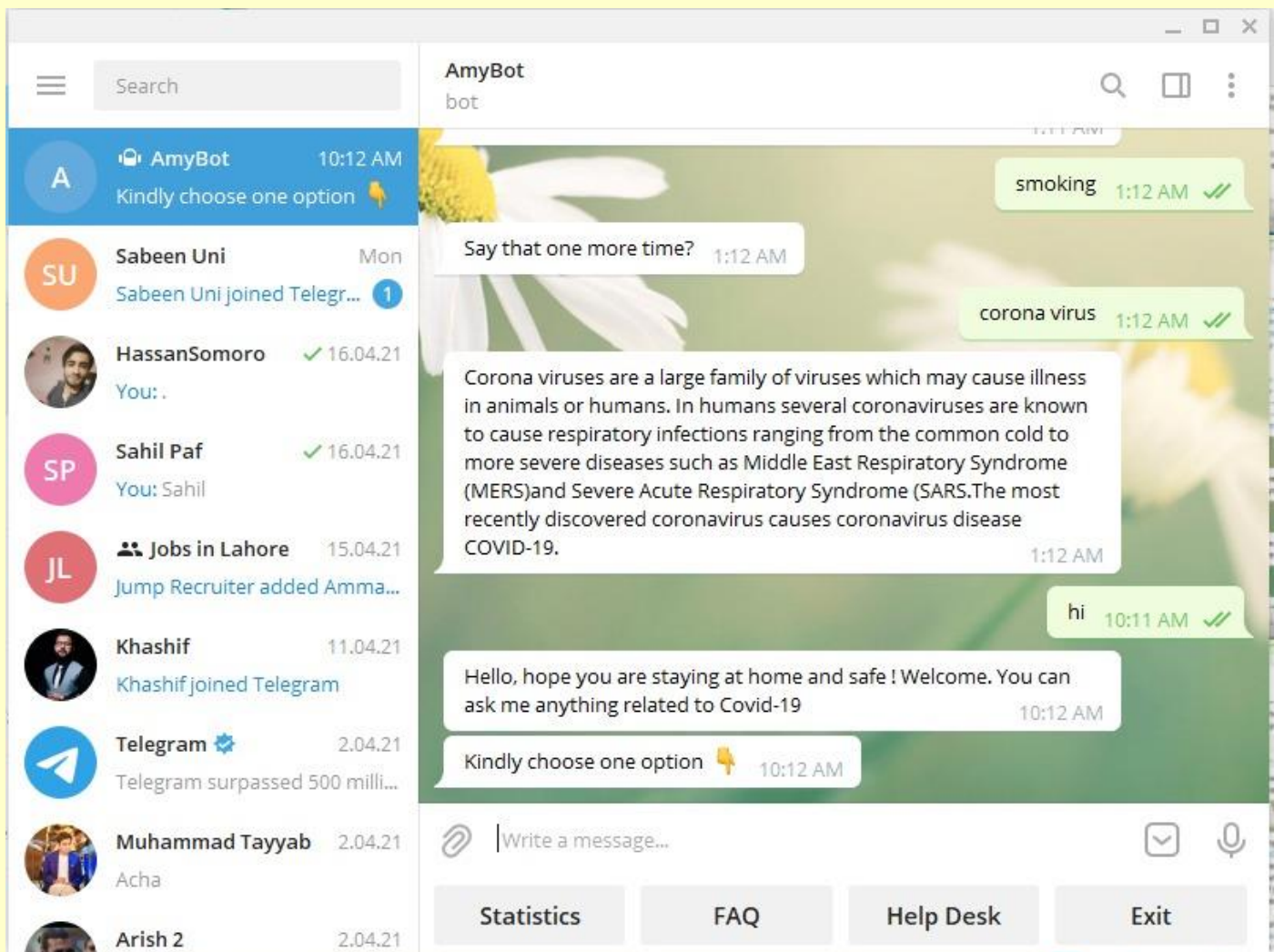
11.0. Creating Chatbot:

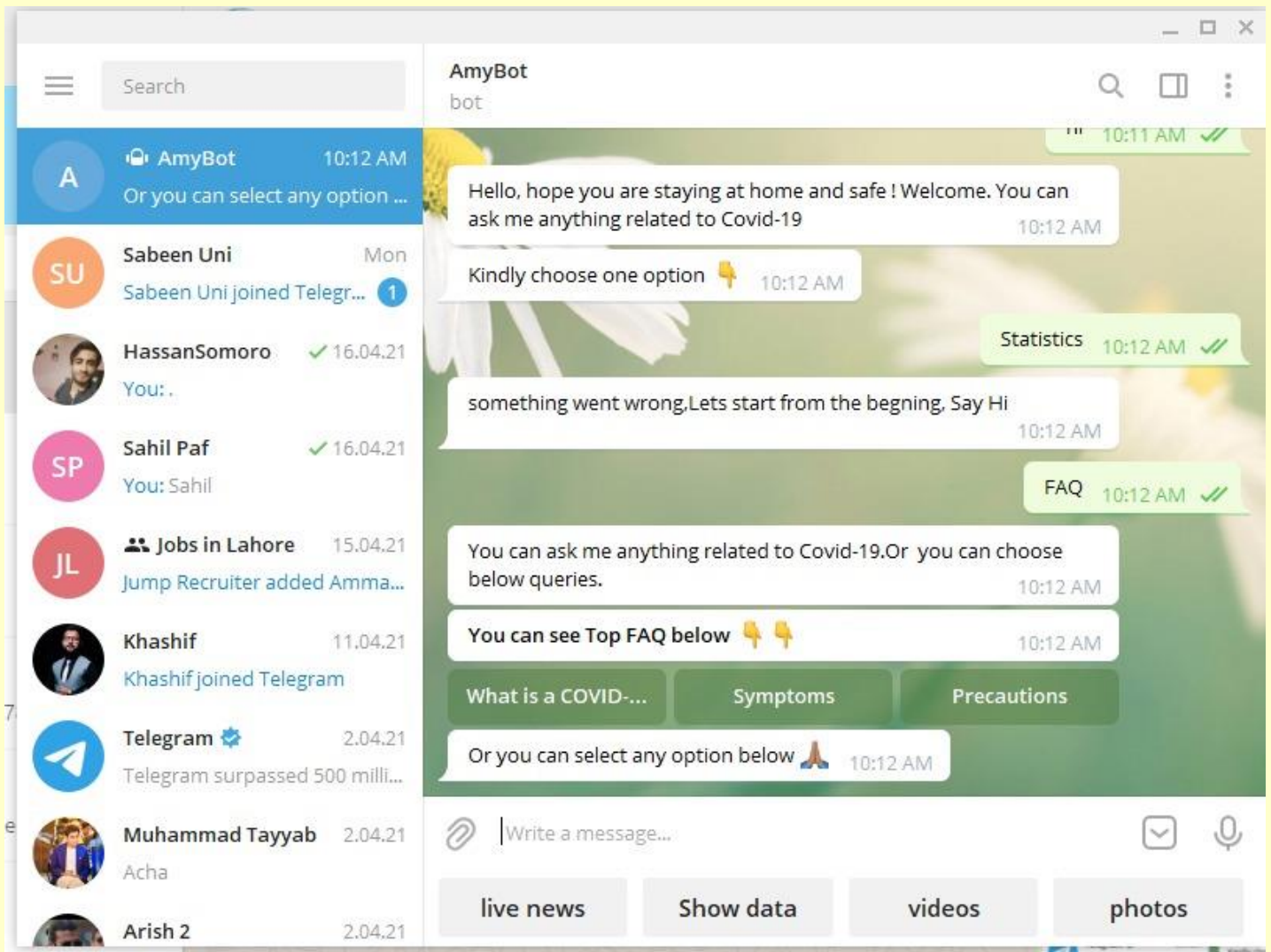


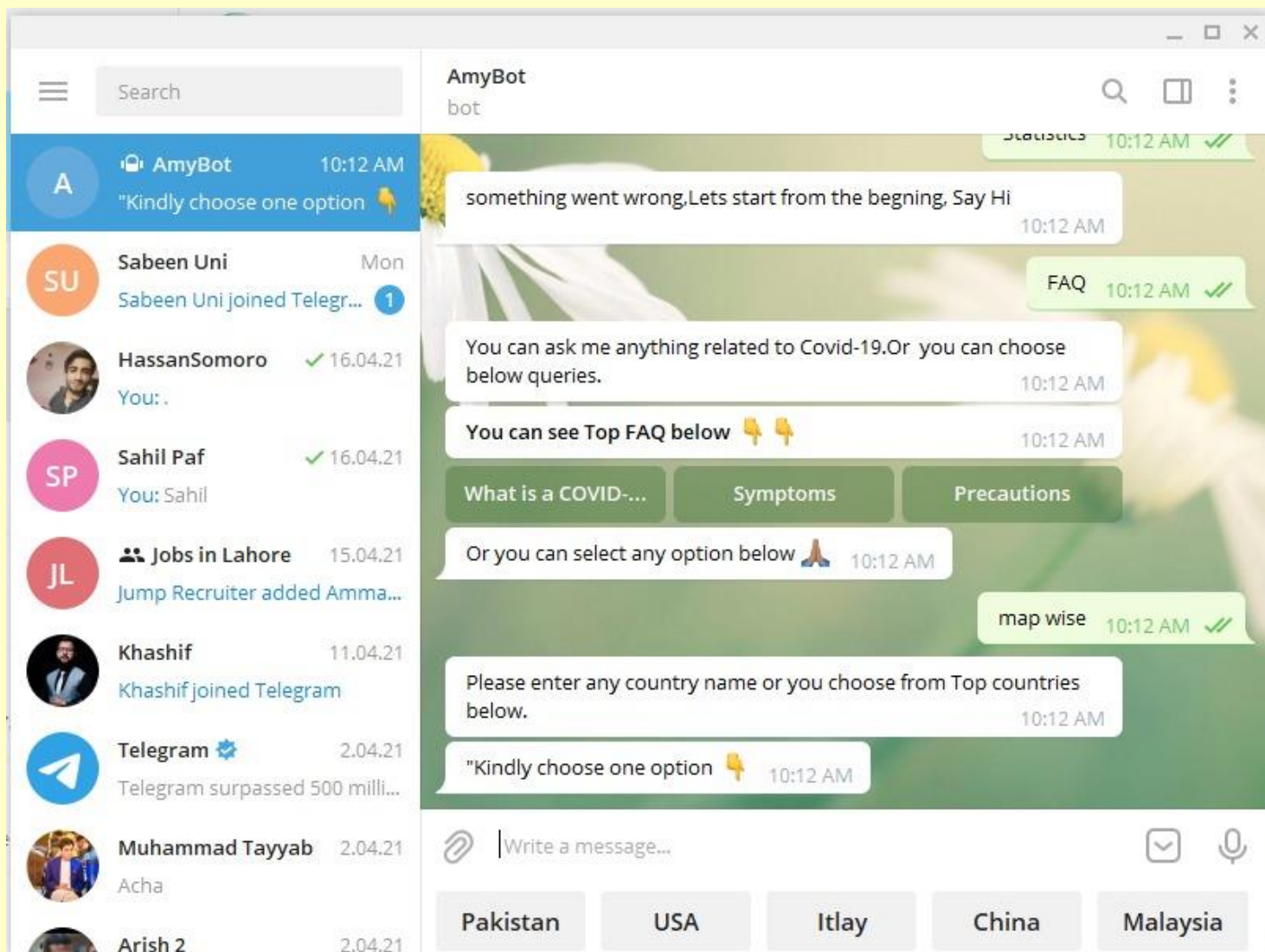


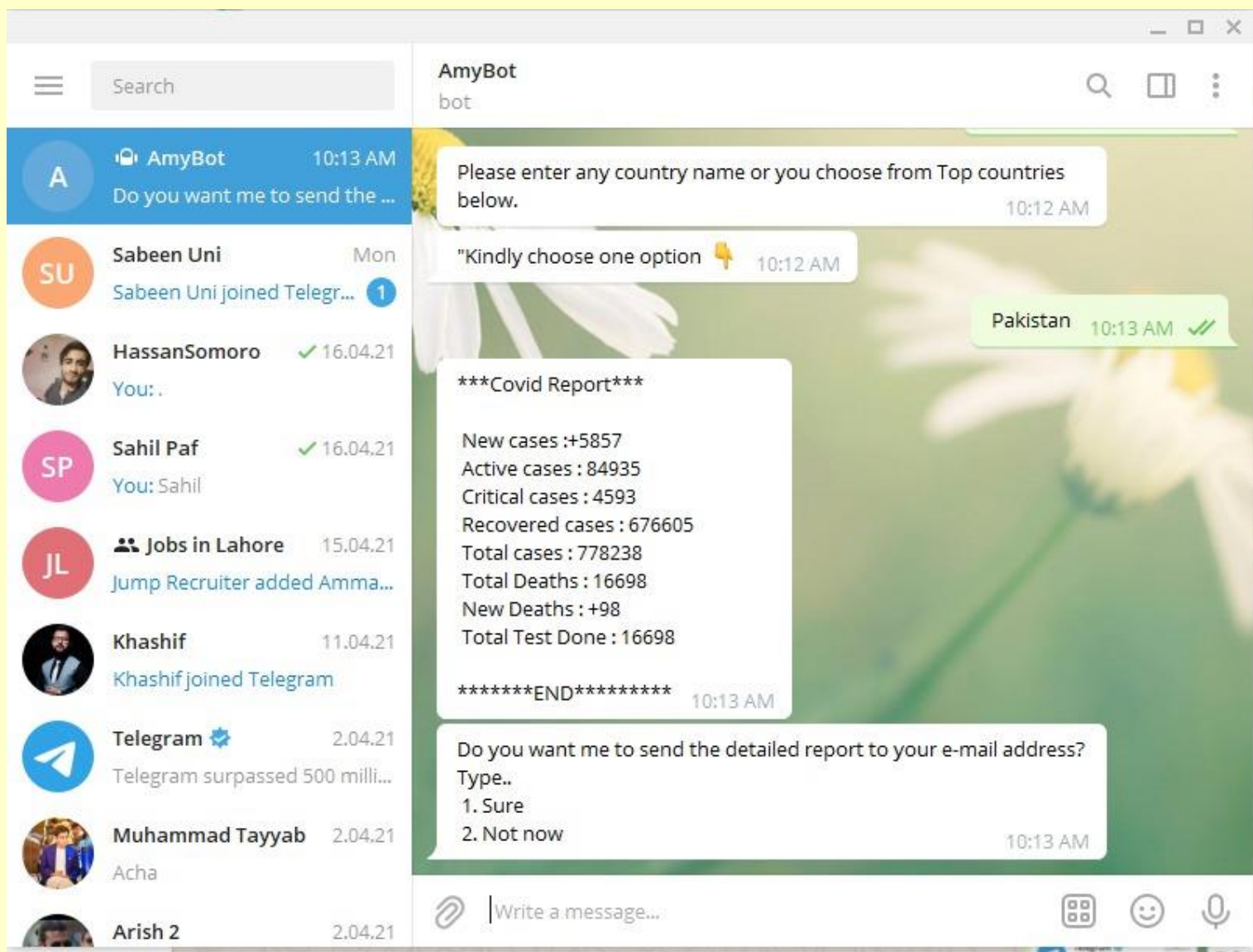


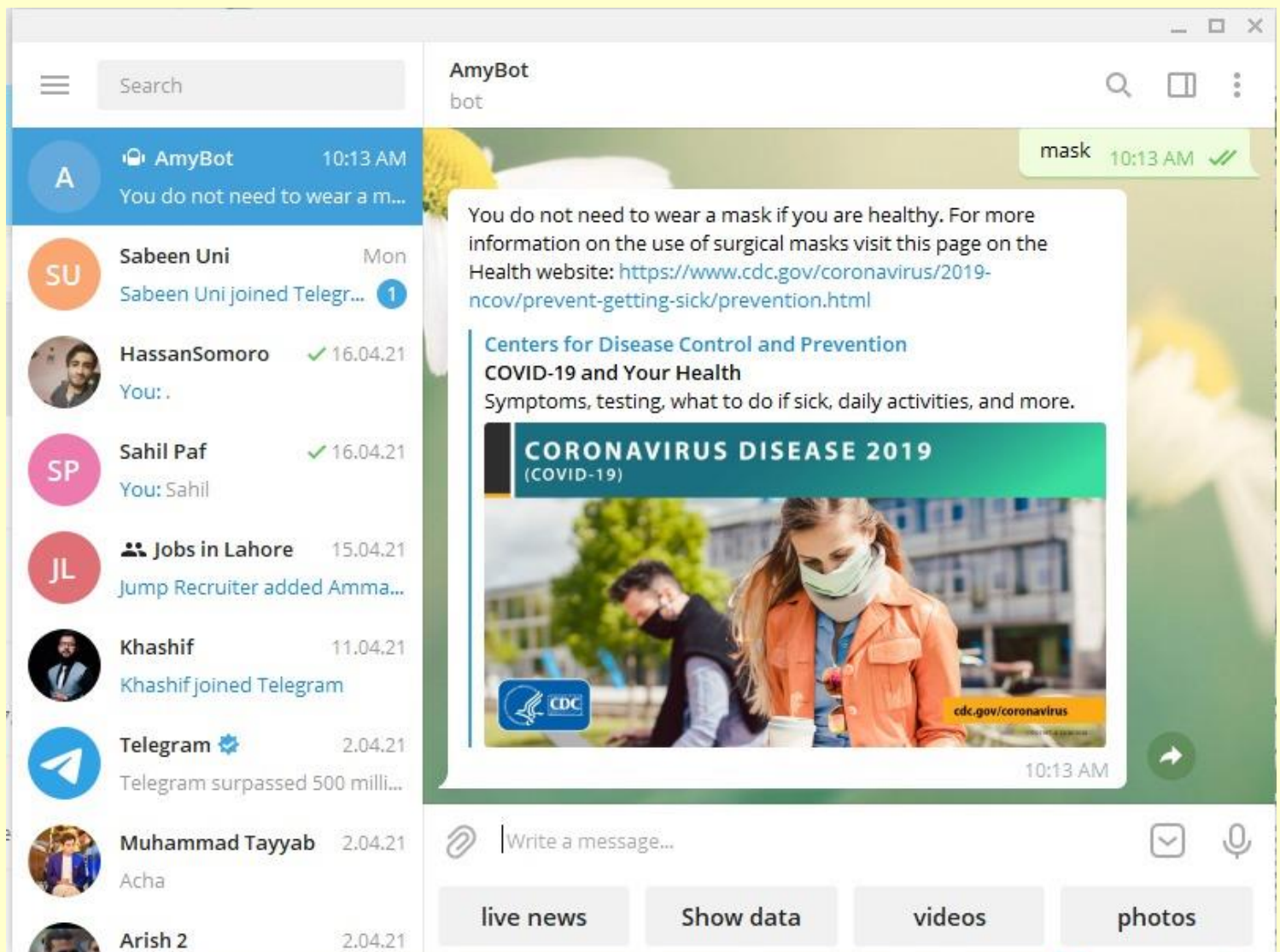
12.0. Output of our project:











Search

A

AmyBot

10:14 AM

here it is. <https://covid.gov.pk/>

SU

Sabeen Uni

Mon

Sabeen Uni joined Telegr... 1

HassanSomoro

✓ 16.04.21

You: .

SP

Sahil Paf

✓ 16.04.21

You: Sahil

JL

Jobs in Lahore

15.04.21

Jump Recruiter added Amma...

Khashif

11.04.21

Khashif joined Telegram

Telegram

2.04.21

Telegram surpassed 500 milli...

Muhammad Tayyab

2.04.21

Acha

Arish 2

2.04.21

AmyBot

bot

Search

COVID-19 and Your Health

Symptoms, testing, what to do if sick, daily activities, and more.

CORONAVIRUS DISEASE 2019

(COVID-19)

10:13 AM

pakistan website

10:14 AM

✓✓

here it is. <https://covid.gov.pk/>

covid.gov.pk

COVID-19 Health Advisory Platform by Ministry of National Health Services Regulations and Coordination

10:14 AM

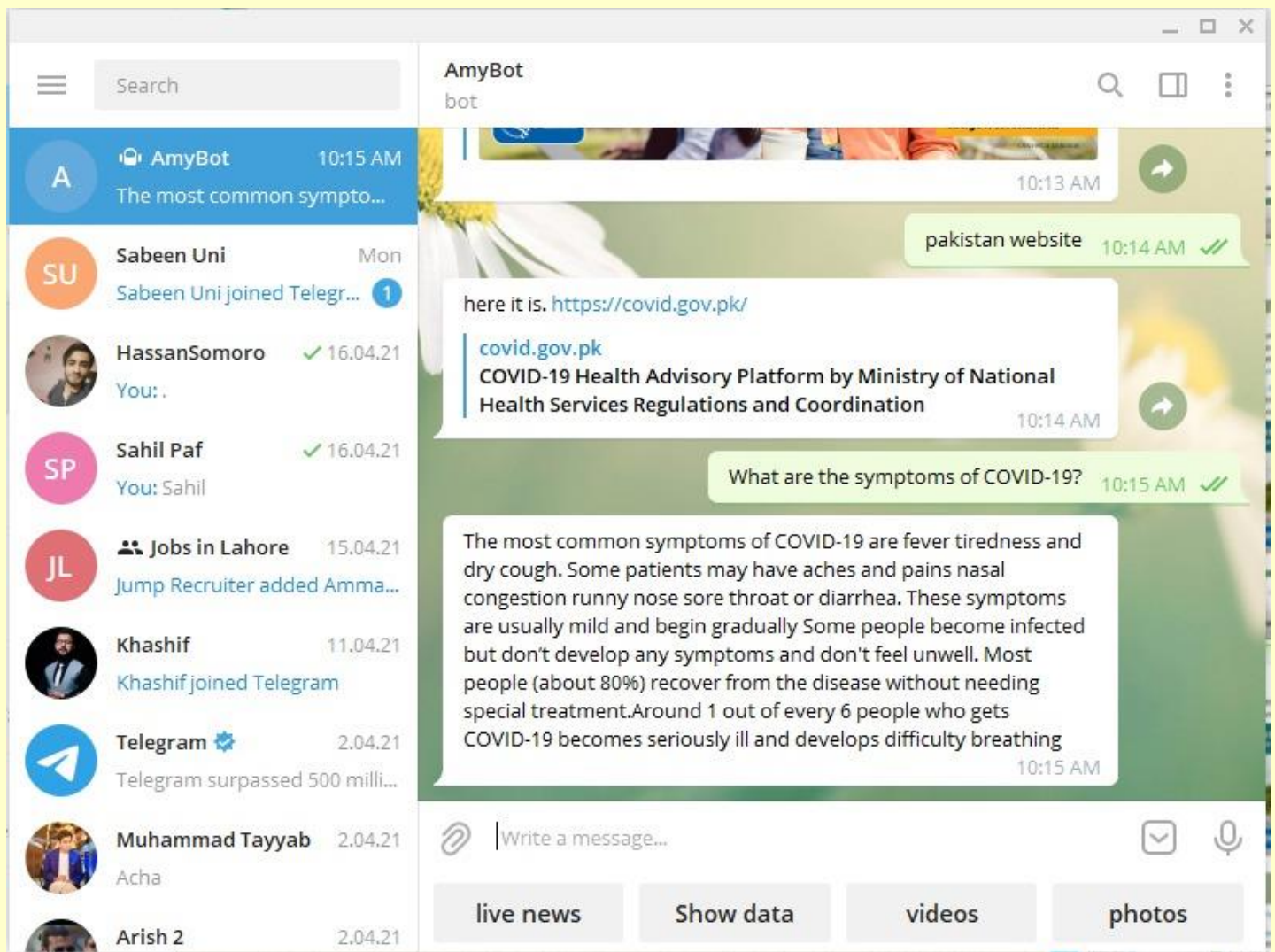
Write a message...

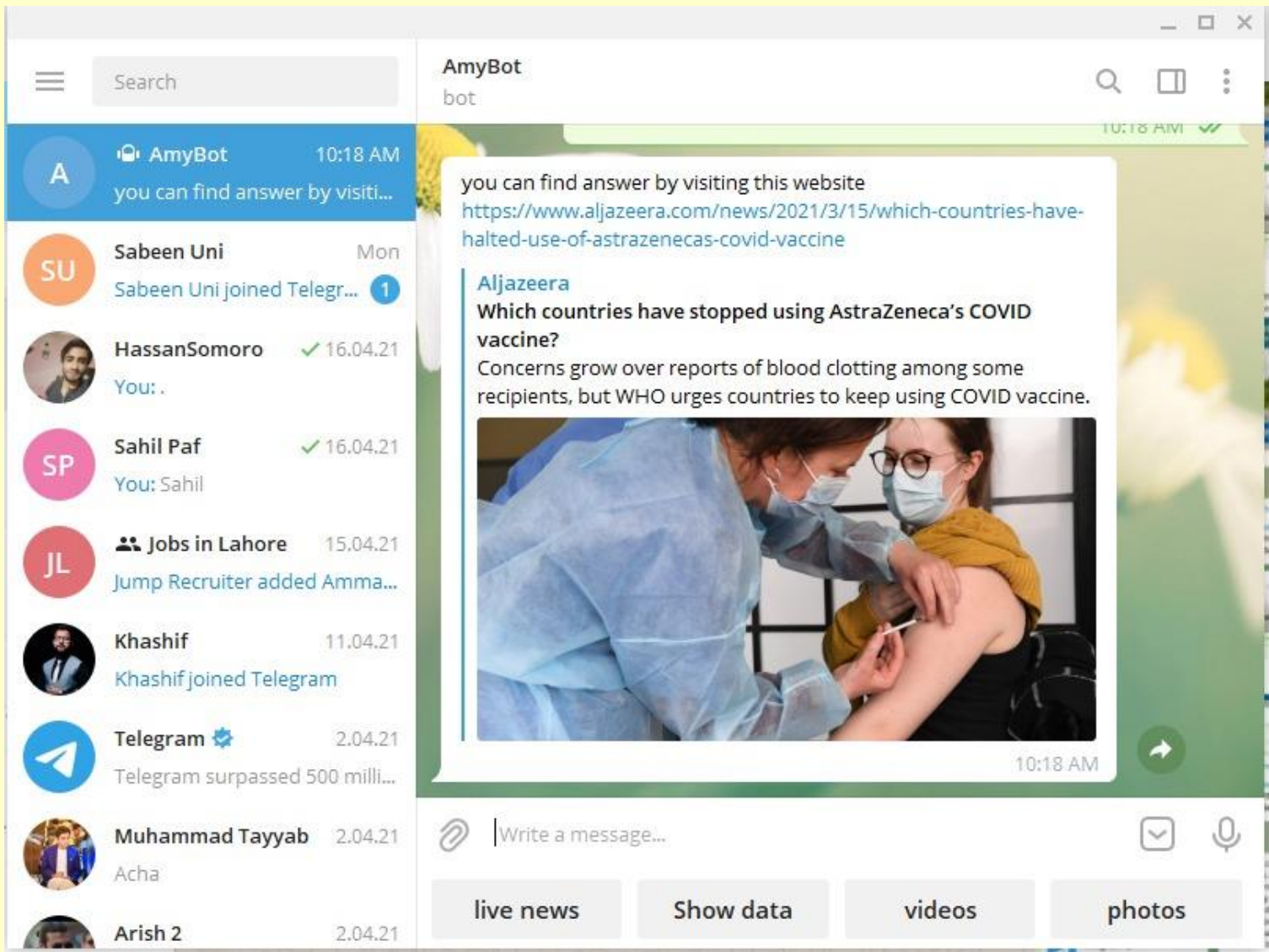
live news

Show data

videos

photos





13.0 Conclusion:

We are all together in a fight against the COVID-19 pandemic. Chatbots, if effectively designed and deployed, could help us by sharing up-to-date information quickly, encouraging desired health impacting behaviours, and lessening the psychological damage caused by fear and isolation. Despite this potential, the risk of amplifying misinformation and the lack of prior effectiveness research is cause for concern. Immediate collaborations between healthcare workers, companies, academics and governments are merited and may aid future pandemic preparedness efforts

