جامعةحلوان
HELWAN UNIVERSITY

كلية الهندسة جامعه حلوان

Factually of Engineering

Computer Engineering Department

Helwan University

MEDIX-E

Health-care System Application

(Corona Virus Detection Unit)

Supervised By

Prof: Ahmed El-Badawi

Prepared By

Khalid Abdul Basset Mohammed

Ziad Ayman Abdul Halim

July-2023

Breakdown of Individual Contributions

Student Information	Contributions
Khalid Abdul Baset Mohamed Section 1	- Design X-ray chest detector - Machine Learning - Back End - Design Lab report detector. - Machine Learning - Back End - Front End - Design Medicine recommender. - Machine Learning - Back End - Front End
Ziad Ayman Abdul Halim Section 2	- Design Android Application. - Design Databases. - Design software Requirements - Deploy machine learning model - Testing

Table Of Contents

- Chapter two: Medicine recommendation system

Abstract

Medix-E is a health care android application, the design is inspired and depending on the imagination of the scenarios that the doctors can rely on to detect any disease.

The application serves the health-care sector via providing multiple features that is necessary to facilitate the patient's care, so the application provides the options of diagnosing the disease from X-ray image that user had done, also the application is able to diagnosing and predicting the diseases from the values of chemical Lab report, and after diagnosing, if the patient was not able to afford cost of buying medicine or the medicine he wants is not available in the market, our application offers a method of medicine recommendations to provide the patients alternatives of his described medicine, all those features are designed very carefully using machine learning techniques to give high accuracy results.

Also, the application offers patients communicate with professional doctors to have a comfortable impression, plus the doctors also able to use our system to have most corrected results to be given to users back.

Keywords

Medix-E, Machine Learning, Image Detection, Data Classification, Recommendation System, Flutter, APIs, Android Application, firebase.

1- General Introduction

In this project, we are going to discuss and implement multiple systems:

Covid-19 detector using X-ray chest image and Covid detector using CBC Lab report and Medicine recommendation system which is able to provide the best five alternatives to the entered medicines and through implementation of each system we will go through multiple stages firstly goes through machine learning sector to build the algorithm that is responsible for prediction, then we will deploy these algorithms to be available for all users, eventually all these features will be combined into android application that will introduce a friendly user interface to make the interactions with these systems is more easy please providing multiple features like communication with doctors.

1.1- Problem statement

Creating a smart system to do the whole functions that doctor does to diagnose any diseases is required because it provides low cost and high accuracy results and simulate all different techniques that doctor takes to diagnosing illness and recommends medicines.

1.2- Objectives

Our objective is to develop a system that must have ability to diagnose the diseases from X-rays chest image and lab report data and the system must be able to also give recommendations of medicine alternatives.

1.3 – Motivations

This application is dedicated to serving the medical field.

By the aid of Machine learning and android we will be able to:

1- Help patients to Diagnosing themselves without need doctor.

2- prevent the spreading of Covid-19 by eliminating interaction human.

3- Reduce the costs of money due to visiting doctors.

4- Achieve high accuracy that doctors cannot provide.

5- Help patients to know and order medicines without needing an expert.

6- help patient to get best alternative medicine suits his budget.

7- help pharmacist to give the client the best alternative medicines.

8- help patient to identify their condition from lab report values.

1.4 – Deployment process

1- machine leering

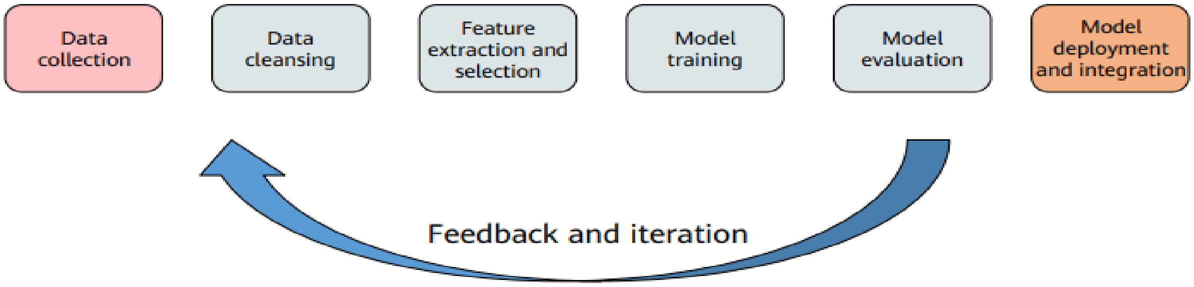


Fig1: Machine Learning process

The operation of developing ML application includes these processes

2- Front-End

3- Back-End

4-Deployment

2- Literature Review

Our system is built with the aid of these researches that had been useful in enhancing the power and smartness of constructed the system.

- COVID-19 Detection: A Systematic Review of Machine and Deep Learning-Based Approaches Utilizing Chest X-Rays and CT Scans.
  - Kirti Raj Bhatele, Anand Jha, Devanshu Tiwari

- COVID-19 Detection on Chest X-ray Images with the Proposed Model Using Artificial Intelligence and Classifiers.

- Muhammed Yildirim, Orkun Eroğlu, Yeşim Eroğlu

- A Literature Review on Recommendation Systems
  - International Research Journal of Engineering and Technology.
  - Shraddha Gupta

- Logistic regression technique for prediction of cardiovascular disease.
  - Ambrish G, Bharathi Ganesh, Anitha Ganesh, Chetana Srinivas.

- Logistic Regression Models in Predicting Heart Disease
  - Yingjie Zhang, Lijuan Diao

3- Analysis and requirements

3.1- System Requirements

3.1.1- Function Requirements

Requirements	Description	Actor
Sign up	Users should create an account to be able to use the application. Both patient and doctor have different data fields in registration.	Doctor & Patient
Log in	The users who have registered are able to login in the application using the email and password to start using the application.	Doctor & Patient
Ask doctor for consultation	The patient can send a request to the doctor for consultation. The patient provides its request with X-ray image, previous diseases, and the existing symptoms.	Patient

Check COVID-19 from X-ray	The doctor receives the patient request and can check the X-ray image for COVID-19.	Doctor
Send diagnosis to a patient	After the doctor checks the patient X-ray image, he can send the diagnosis to the patient. The diagnosis consists of the result of COVID-19 check, the doctor advice and the recommended drugs.	Doctor
Make private checks	The doctor can upload an image for the X-ray from his mobile or from camera to check it. he can write notes on this check and save the result in his private diagnoses.	Doctor
Update the profile	Both doctor and patient can update their profile.	Doctor & Patient

Having Medicine recommendations	Both doctor and patient can use the system to have the best 5 alternatives of a certain medicine, in case of lacking it in the market	Doctor & Patient
Check covid-19 from Lab Report	Ability to check the disease from the critical values that related to the disease.	Patient

3.1.2- NONE-Function Requirements

Requirements	**Description**
Availability	The app is available all the time whenever there is an internet connection.
Usability	It is very easy to understand the flow of the app and use to use it.
Portability	Our mobile app anyone can install it on his mobile and use it.
Reliability	All system functions work well and there are no fails.
Security	The System provides a good level of privacy for all users’ data.

3.1.3- Functional Requirements Specifications

- **Stakeholders**:

- Doctor
- Patient

- **Actors and goals:**

- Patient
  - Choose a doctor with a specialize that he wants and sends him his condition providing it with X-ray image.
  - Having medicine recommendation system to get alternatives.
  - Able to predict the disease he had via lap report values
- Doctor
  - Receive the request from the patient and make check on the X-ray image and send the response to the patient consisting of COVID-19 check result,
  - Having medicine recommendation system to get alternatives. and advice to the patient and recommended drugs.

Use Cases:

Use Case Description:

Register	Users should create an account to be able to use the application. ⇨ **Requirements**: inserting the required credentials
Login	To make authentication to the system, the user should enter the amil and password. **Requirements**: email and password.

Send request to a doctor	The patient sends a request to a doctor to diagnose it. **Requirements:** X-ray image, symptoms, and any previous diseases.
Get Covid-19 prediction	The doctor uses the ML model to check the incoming image from the patient. **Requirements:** the incoming X-ray image
Send a diagnosis to a patient	After the doctor gets the result from the model, he writes his diagnosis and sends all that to the patient. **Requirements:** login, receive the patient request, write advice and recommended drugs to the patient and the result of the COVID-19 check
Make private diagnosis	The doctor can upload images and use the model to check these images and write notes on them and then save them. **Requirements:** login, upload image.
Make medicine recommend	The doctor and patients can have the best 5 alternatives of medicine. **Requirements:** writing the name of medicine.
Lab predicts	The patient can predict the disease from CBC values

➢ Use case diagram:

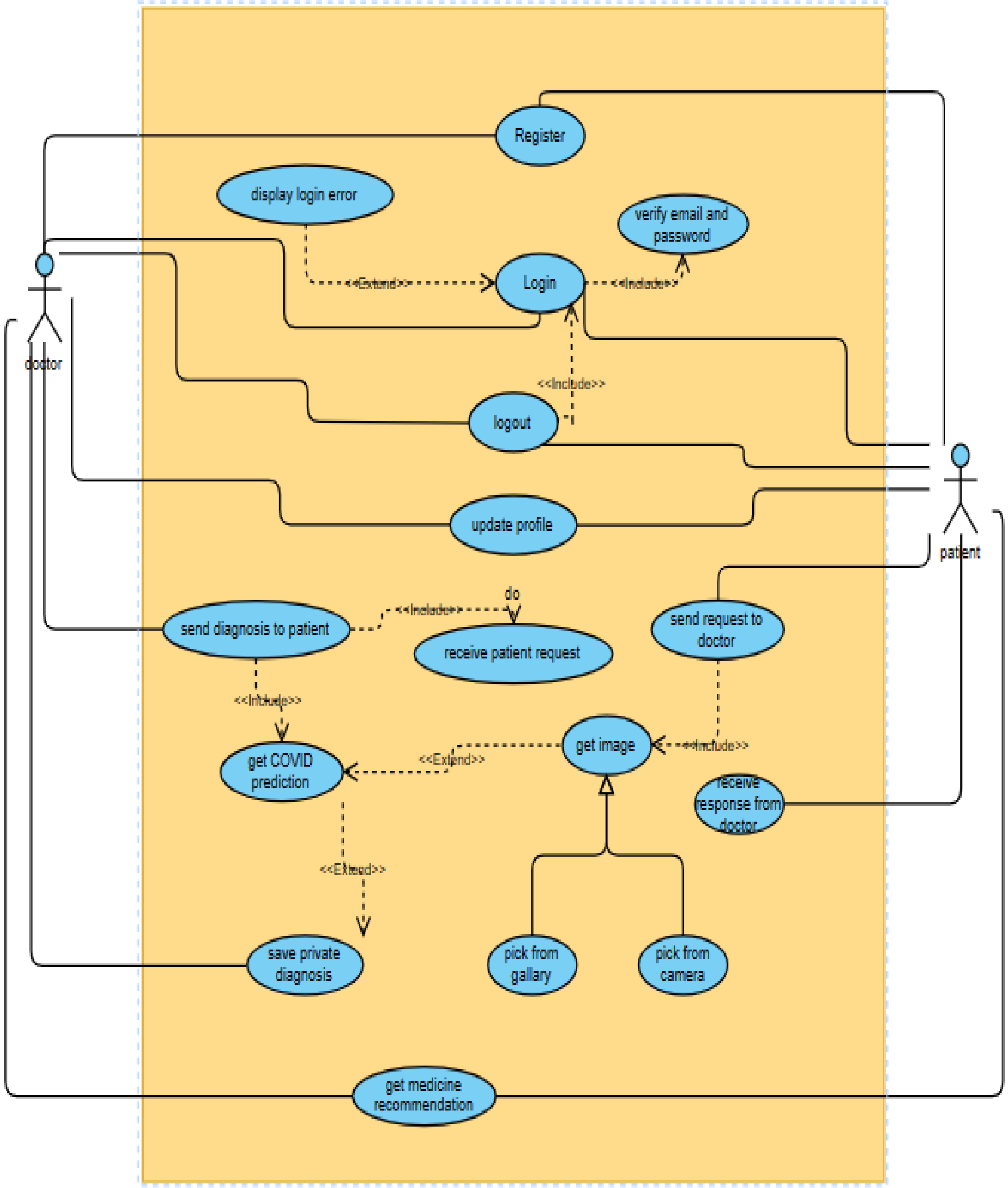


Fig2: Use Case Diagram

➢ **Sequence diagram:** **1. patient register**

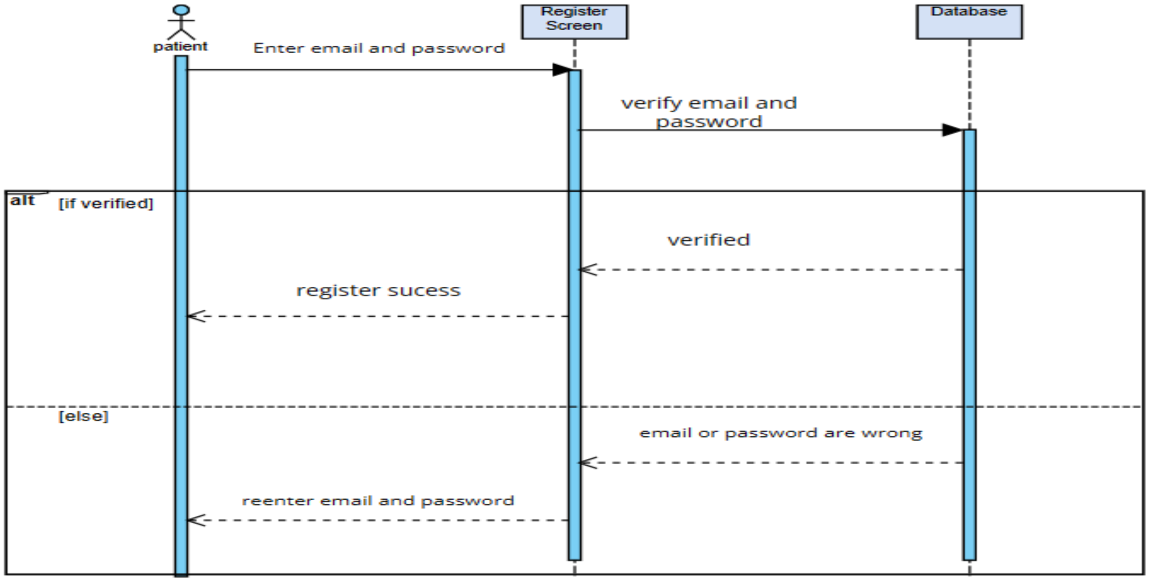


Fig3: Sequence Diagram: patient register

2. patient login

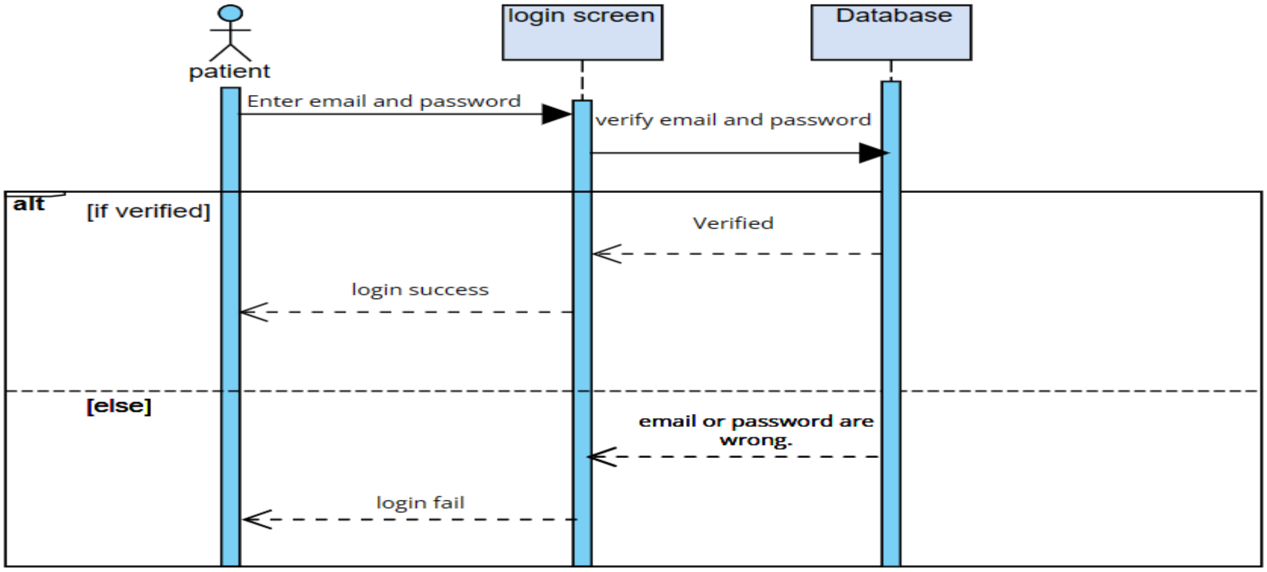


Fig4: Sequence Diagram: Patient Login

3. doctor register

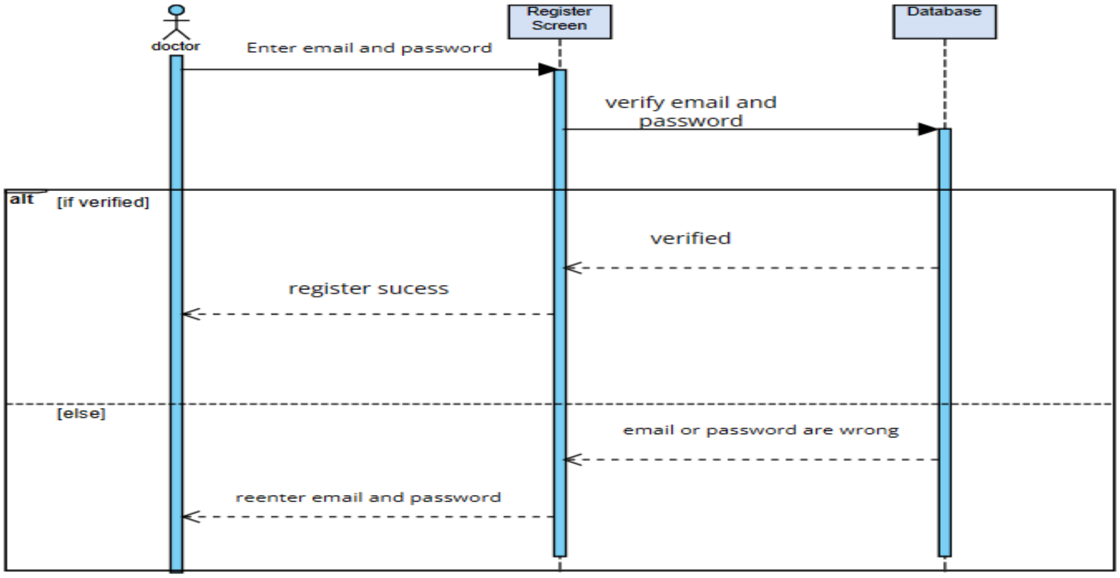


Fig5: Sequence Diagram: Doctor register

4.doctor login

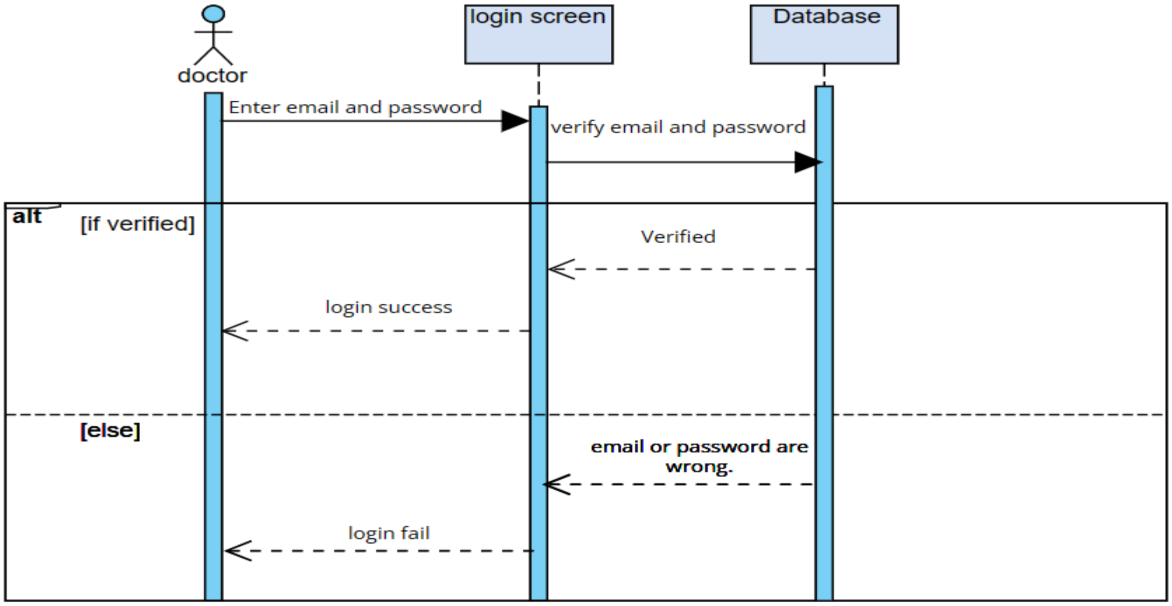


Fig6: Sequence Diagram: Doctor Login

5. Communication between the doctor and the patient

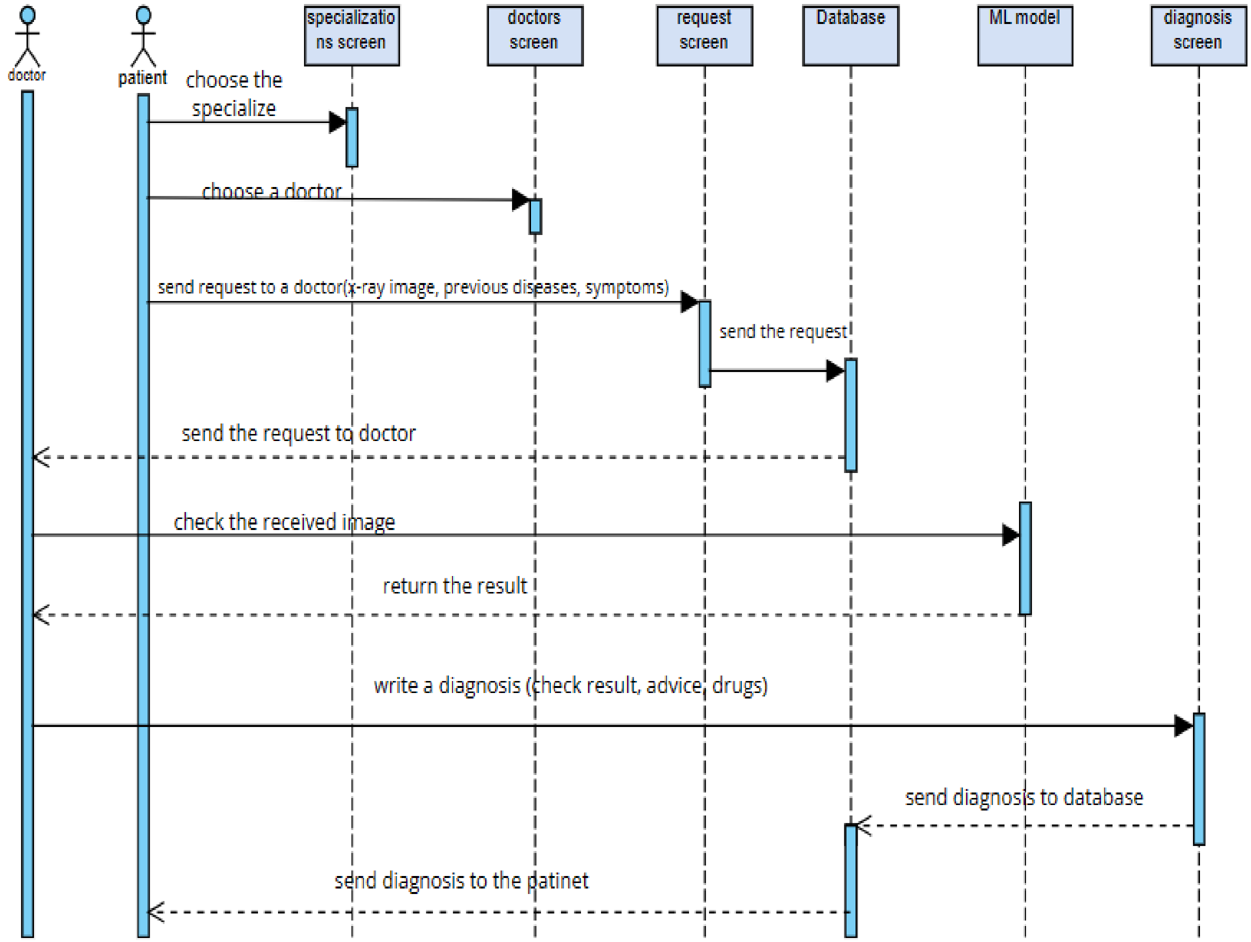

doctor
patient
specializations screen
doctors screen
request screen
Database
ML model
diagnosis screen
choose the specialize
choose a doctor
send request to a doctor(x-ray image, previous diseases, symptoms)
send the request
send the request to doctor
check the received image
return the result
write a diagnosis (check result, advice, drugs)
send diagnosis to database
send diagnosis to the patinet

Fig7: Sequence Diagram: Patient doctor Communication

➢ System Architecture

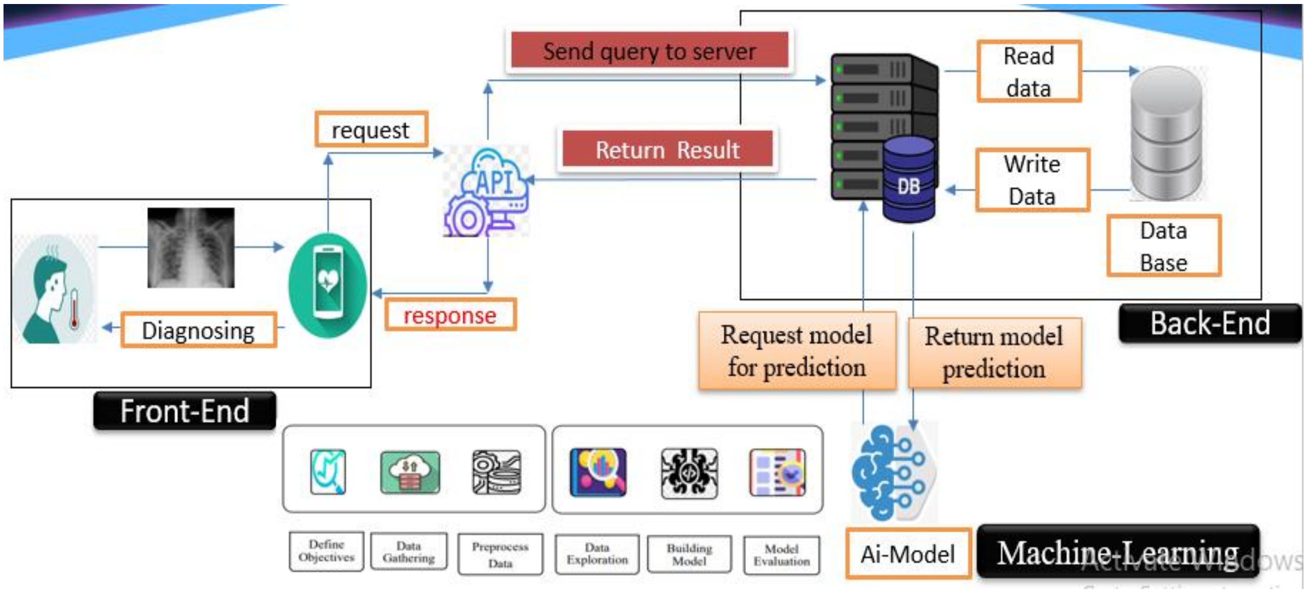
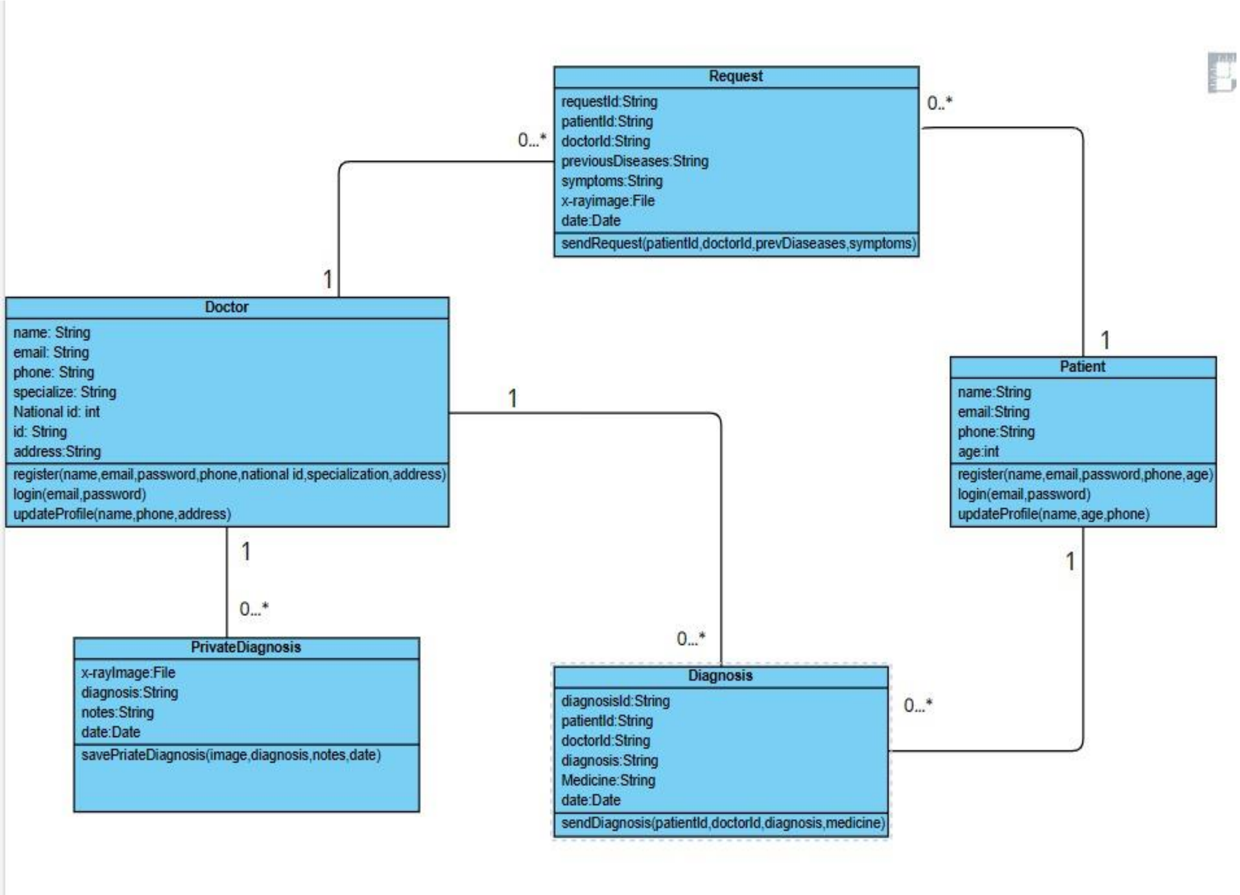


Fig8: System Architecture

➢ Class Diagram



4- Technology and Tools

Technologies and programming languages:

- Machine Learning: We will use several ML models like KNN and Random Forest algorithms.

- Flutter: flutter will help us to build our cross platform mobile application.

- Pandas: is a tool for data analysis, which is a Python library.

- Flask framework: will help us to build our API that will connect to the mobile app.

- Python: the programming language that will be used for data analysis, building our machine learning model, and building our API.

- Dart: the programming language that will be used for building our mobile application.

Tools:

- Android Studio: the software that will be used for building the app.

- Anaconda: the software that will be used for making data science and machine learning works.

- Streamlit and render is a free server that we will use for uploading our API on it.

- spyder: is a program used for data splitting and saving them as train and test files.

5- Implementation

Chapter one: X-rays (Covid-19) Detector

1- Introduction: -

In this application, we are going to use the image classification process techniques to detect whether the chest x-ray image that user gave is normal or it's abnormal (Covid-19).

That will spare the patient from going or waiting to be diagnosed.

The application is represented into mobile application and enable user to do self-examination or sending the image to professional person that also capable to use the benefit of the artificial intelligence image detection.

1.1- Problem Statement: -

Despite the availability of several diagnostic methods for COVID-19, such as RT-PCR and antigen tests, there is still a growing need for an efficient and accurate tool for detecting the disease, particularly in resource-limited settings, X-ray imaging is a useful modality for detecting COVID-19, as it can reveal characteristic patterns in the lungs of infected individuals. However, accurate interpretation of X-ray images requires specialized training and expertise.

1.2- Objectives: -

1- Gathering a collection of normal and (covid-19) images as dataset

2- Preprocessing these images by filtering, zooming, feature extraction

3- Develop a machine learning model, (CNN), that can classify a given X-ray image as COVID-19 positive, negative, or inconclusive.

4- Evaluate the performance of the model.

5- Develop a user-friendly interface that allows clients to use models.

1.3- Motivations: -

This application is dedicated to serving the medical field.

1- Helps patients to Diagnosing themselves without need doctor

2- prevent the spreading of Covid-19 by eliminating the interaction between patient and doctor.

3- Reduce the costs of money due to visiting doctors.

4- Achieve high accuracy that doctors cannot provide.

1.4- Development Operation: -

1- machine leering

2- Front-End: will be discussed in android chapter.

3- Back-End: will be discussed in android chapter.

4-Deployment: will be discussed in android chapter.

2- Implementations: -

2.1- Machine Learning

2.1.1- Data Collection: -

2.1.1.1- Covid-chest Xray-dataset: -

- **Name of researcher:** Dr/ joseph Cohen
- **Specifications of the dataset:**
- This dataset is a collection of 930 chest Xray images.
- Those images represent several diseases: Covid-19, SARS,..etc.
- It also contains multi-sectional view images.

Why this data set must be preprocessed?

1- selecting the chest of (+) covid-19 patients among other diseases

2- selecting the PA view among the other selection views.

2.1.1.2- Chest X-Ray Images (normal) dataset

- Name of researcher: Paul moony
- Specifications of the dataset:
- This dataset is a collection of 1341 chest Xray images.
- These images represent normal chest Xray images.

2.1.2- Data Preprocessing (Cleansing): -

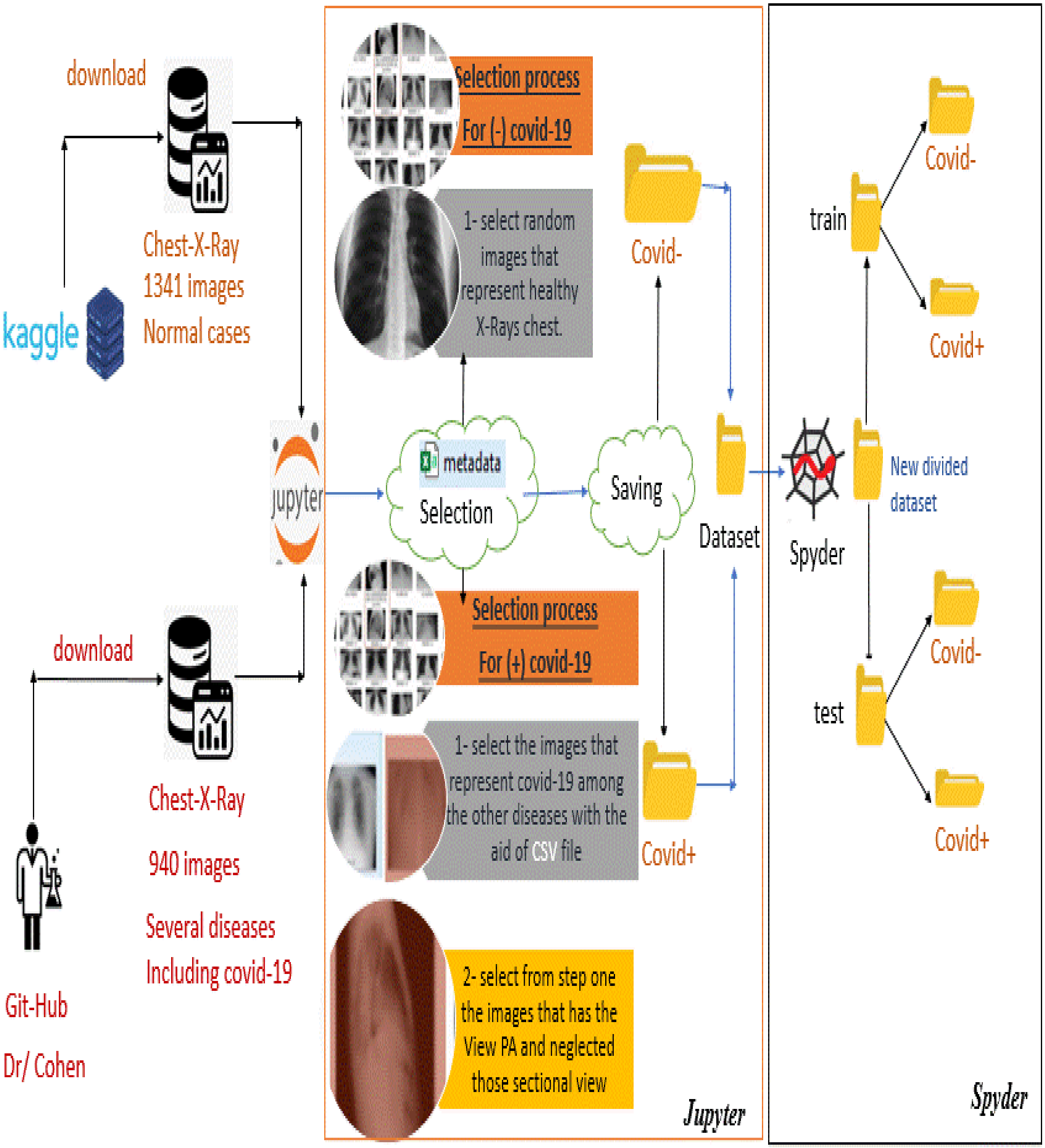


Fig9: (Covid-19-Xrays) preprocessing Data set.

2.1.2.1-Image Selection: -

1- Introduction

Due to the plenty of unnecessary images (sectional images,other diesaeses..etc)

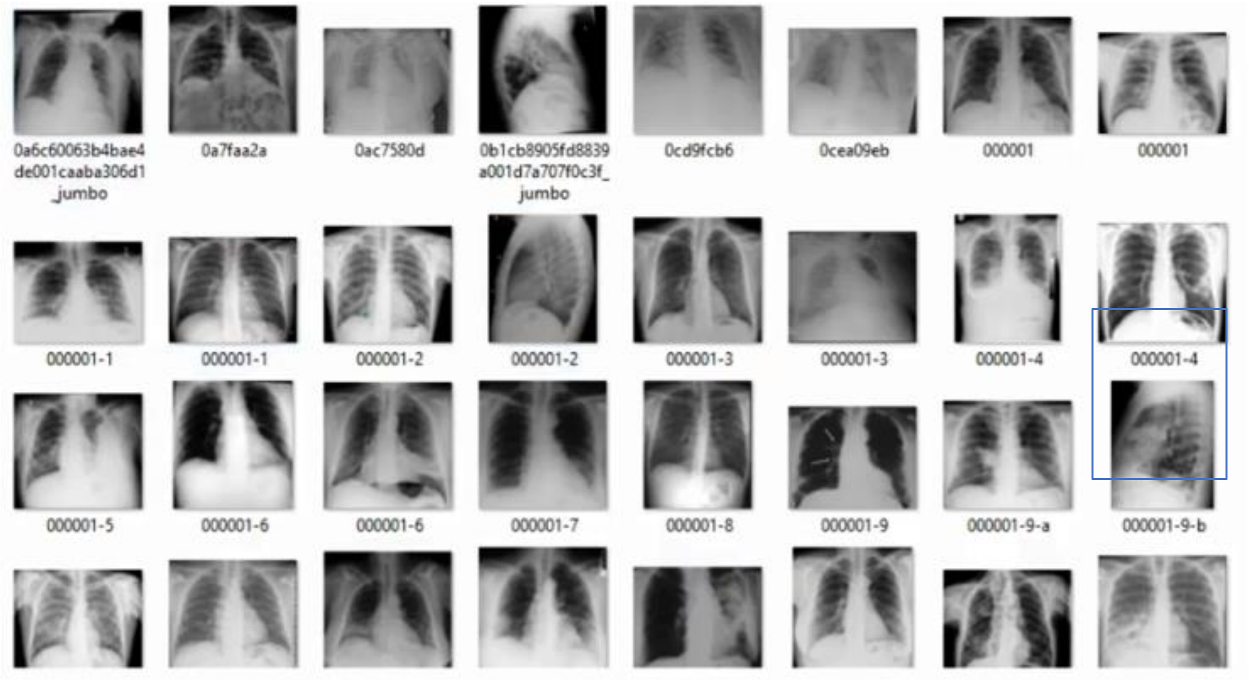


Fig10: (Covid-19-Xrays) before image selection.

we need to preprocessing the dataset to select the images

2-Technique of Image selection:-

2-1 Problem:-

The tradiational (manual) ways of selection the needed images and get rid of the unneeded images are not effective because:

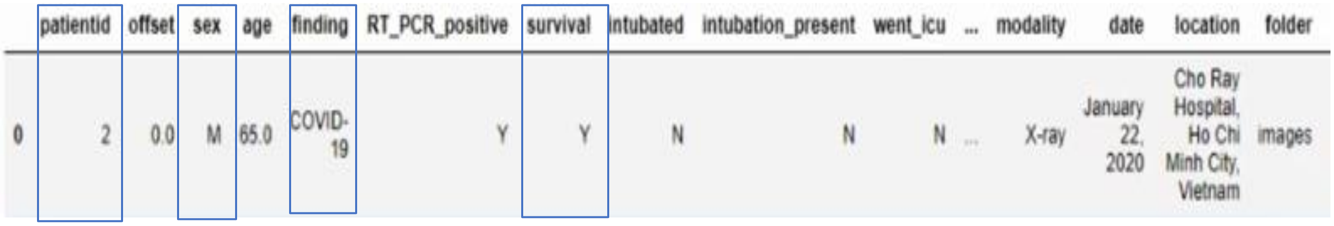
- not practical
- takes a lot of time and effort
- tedious
- not exclusive

2-2 Solution:-

We will deal with the CSV file (METADATA) of the dataset and through it, we can manupulate, search and select the images which matches the desired spesifications and copy those images into their own folder for future use.

CSV file (metadata)

- **it's the file that contains the whole information about the dataset**
- **those info could be:** age of patient, modilty of image, sex, id,....etc
- **the Info help in:** selection specific images among multiple images



	patientid	offset	sex	age	finding	RT_PCR_positive	survival	intubated	intubation_present	went_icu	...	modality	date	location	folder
0	2	0.0	M	65.0	COVID-19	Y	Y	N	N	N	...	X-ray	January 22, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images

Fig11: (X-rays-data set) Metadata file.

2-3 The operation of Selection specific images among other:-

We aim to select the images that

- represent covid-19
- their type PA view (front to end projection)

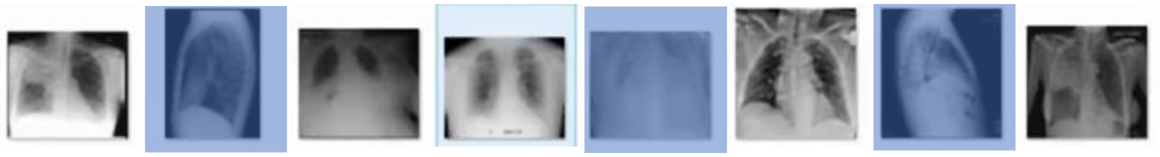
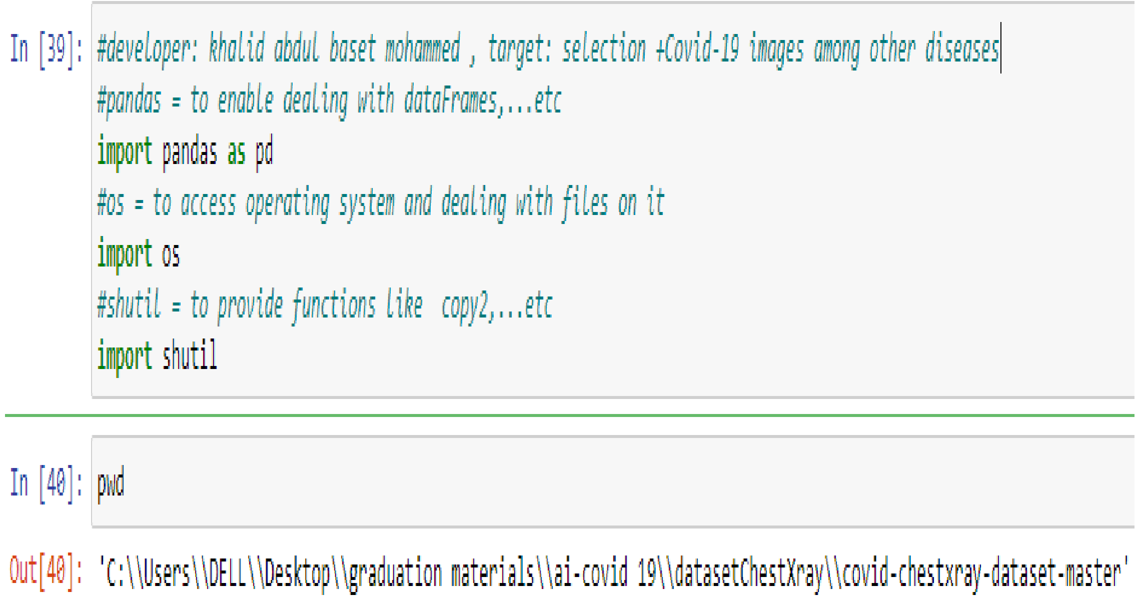


Fig12: (X-rays-data set) during image selection.

2-4 Python Code for preparing the + covid-19 dataset:-

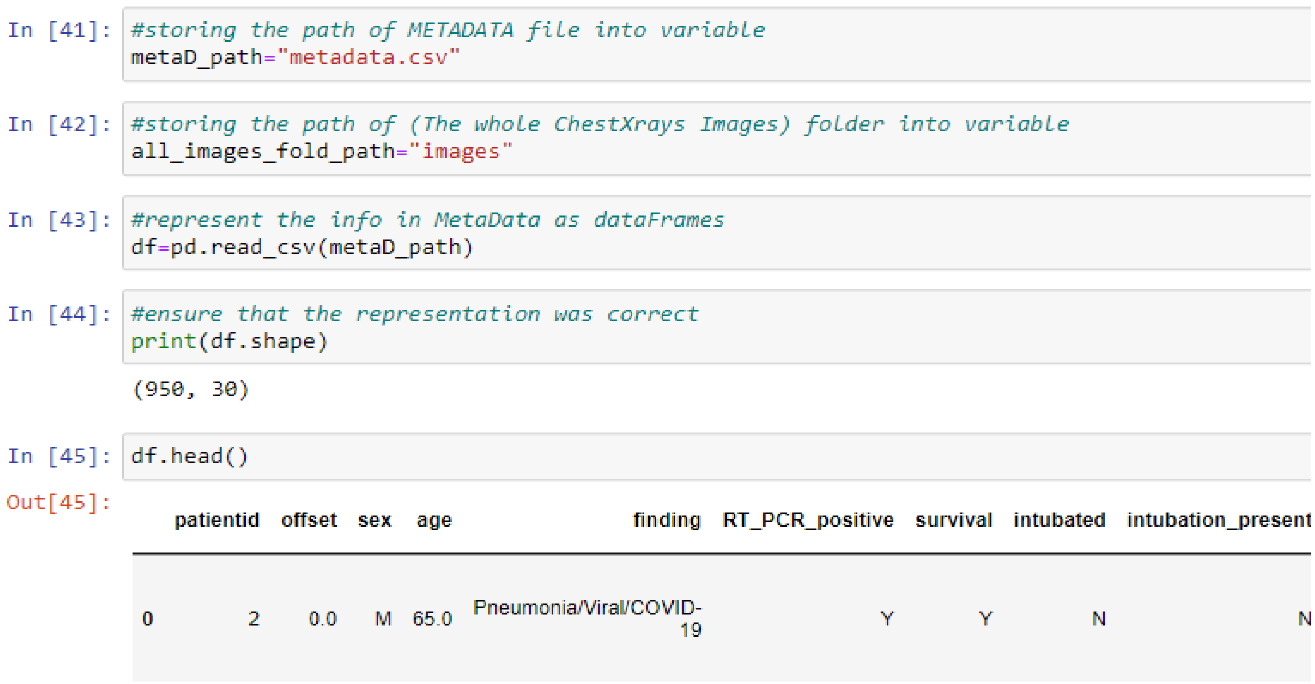


```
In [39]: #developer: khalid abdul baset mohammed , target: selection +Covid-19 images among other diseases
         #pandas = to enable dealing with dataFrames,...etc
         import pandas as pd
         #os = to access operating system and dealing with files on it
         import os
         #shutil = to provide functions like  copy2,...etc
         import shutil

In [40]: pwd

Out[40]: 'C:\\Users\\DELL\\Desktop\\graduation materials\\ai-covid 19\\datasetChestXray\\covid-chestxray-dataset-master'
```

By import the folder that contains (CSV file, Images file, etc.) to Jupyter

1- importing

pandas	Enable the dealing with data as structure & supply functions
os	Enable interaction of user with native operating system functions
shutil	Offers a number of operations on files and collections of files
pwd	To know where the directory path that i am standing currently



```
In [41]: #storing the path of METADATA file into variable
         metaD_path="metadata.csv"

In [42]: #storing the path of (The whole ChestXrays Images) folder into variable
         all_images_fold_path="images"

In [43]: #represent the info in MetaData as dataFrames
         df=pd.read_csv(metaD_path)

In [44]: #ensure that the representation was correct
         print(df.shape)

         (950, 30)

In [45]: df.head()
```

Out[45]:

	patientid	offset	sex	age	finding	RT_PCR_positive	survival	intubated	intubation_present
0	2	0.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N

2- storing in variables

MetaD_path	Store the path directory of the CSV file into this variable
all_images_fold_path	Store the path directory of the folder that contains all unclassified Images into this variable

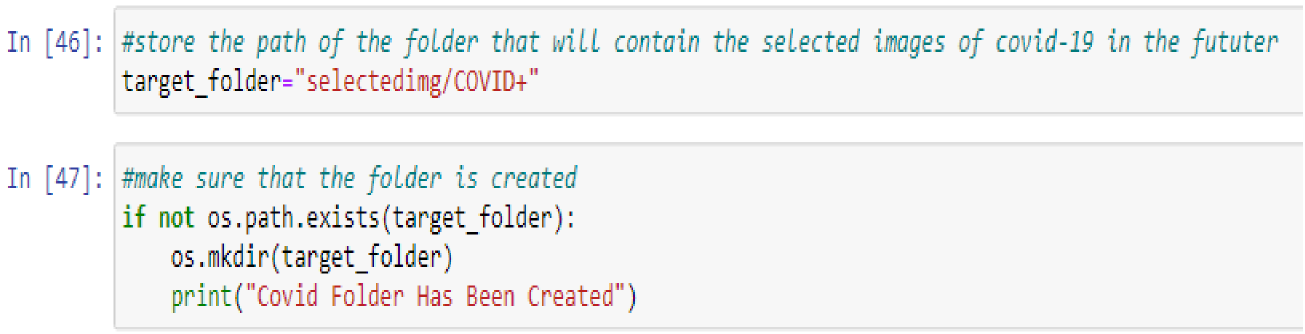
3- represent the CSV file as data Frame Structure

df=pd.read_csv(file_path)	Store the info of the csv file as data frames so that it can be easy to deal with them

4- display the created data frames' specifications

Print(df.shape)
df.head()
	It turns out that our data set contains 950 images each has 30 information like(age, sex , finding(o/p),…etc)

5- Selecting the images and copy them

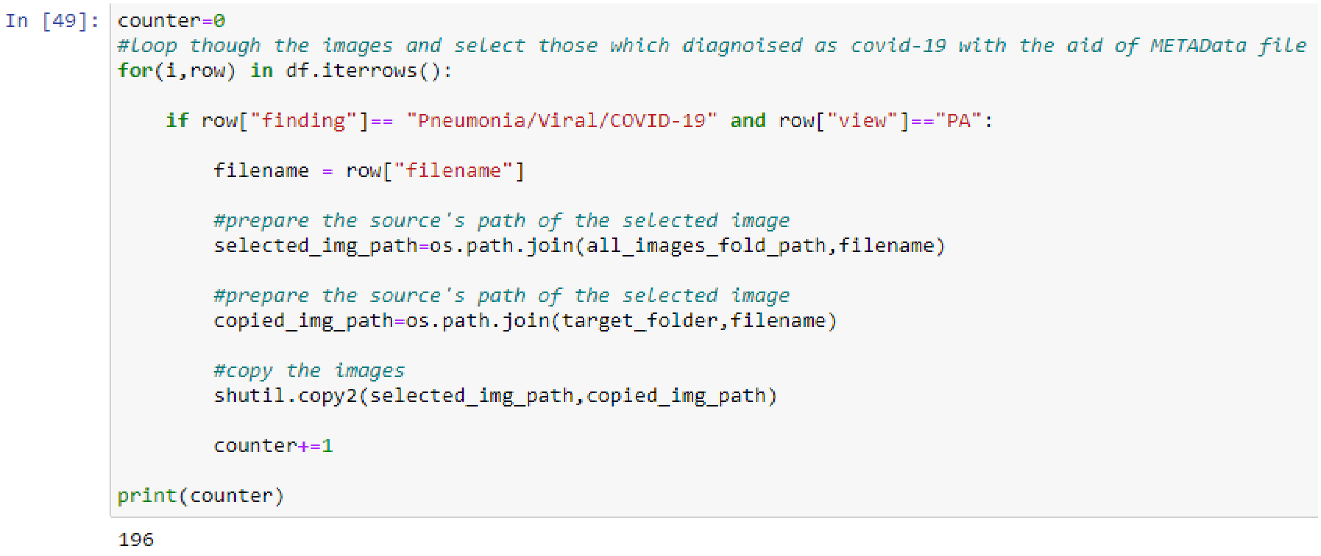


```
In [46]: #store the path of the folder that will contain the selected images of covid-19 in the fututer
         target_folder="selectedimg/COVID+"
```

```
In [47]: #make sure that the folder is created
         if not os.path.exists(target_folder):
             os.mkdir(target_folder)
             print("Covid Folder Has Been Created")
```

target_folder=”selecting /COVID+”	Variable that contains a directory of the folder of the file Covid-19 which will contains the covid-19 images only later.

If not os.path.exists(target_fold):
Os.mkdir(target_fold)
Print(“Covid folder created”)
	1- check if the OS does not contain the file
	2- ask os to create one
	3- print an sentence to ensure the creation



```
In [49]: counter=0
         #loop though the images and select those which diagnoised as covid-19 with the aid of METAData file
         for(i,row) in df.iterrows():

             if row["finding"]== "Pneumonia/Viral/COVID-19" and row["view"]=="PA":

                 filename = row["filename"]

                 #prepare the source's path of the selected image
                 selected_img_path=os.path.join(all_images_fold_path,filename)

                 #prepare the source's path of the selected image
                 copied_img_path=os.path.join(target_folder,filename)

                 #copy the images
                 shutil.copy2(selected_img_path,copied_img_path)

                 counter+=1

         print(counter)

         196
```

1-Counter=0 2-For(i, row) in df.iterrows(): 3-If row[“finding”]==”covid-19” and row[“view”]==”PA”: 4-filename=row[”filename”] s_img_p=os.path.join(images_path,filename) c_img_p=os.path.join(target_fold, filename) 7- shutil.copy2(img_p, img_cp_p) 8- counter++ 9-print(counter)	1- create a counter to indicate the number of selected imgs 2- create a loop to search over all images’ specifications 3- searching for the images that diagnosed as covid19 and viewed as PA 4- copy the file name of this image into var calls file name 5- prepare source directory of selected image 6- prepare the destination directory of selected image “NEW FOLDER”

6- Conclusion: -

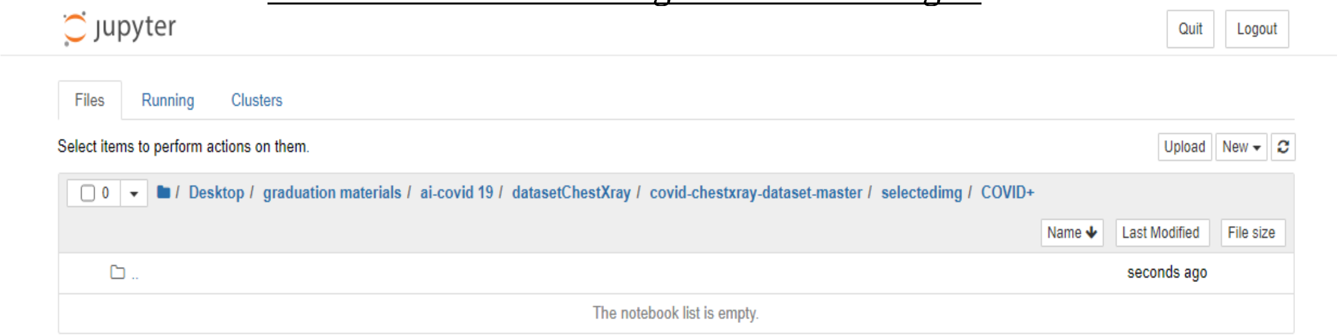
Conclusion

Among 950 Chest-Xray with different views and diagnosis.: -

1- select 196 Chest-Xray diagnosed as covid and viewed as PA

2- save them into a folder to use them for building the model and train it.

The folder before selecting the + Covid images.



Select items to perform actions on them.

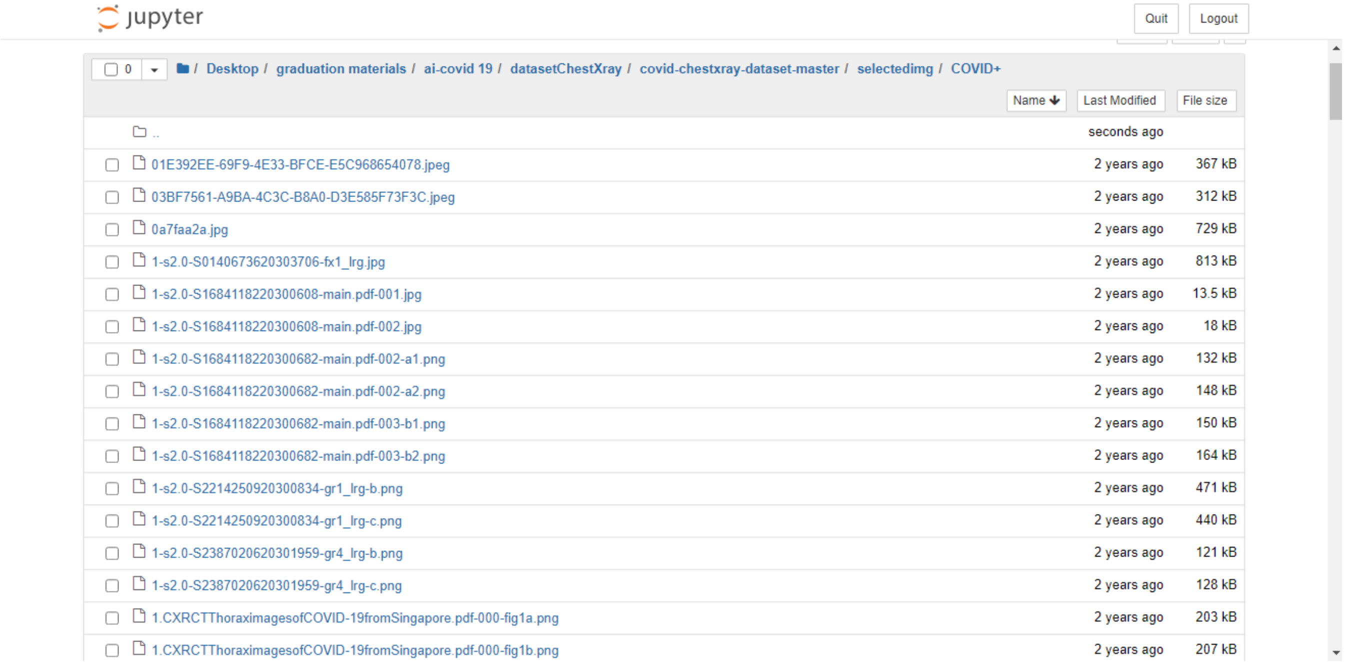
/ Desktop / graduation materials / ai-covid 19 / datasetChestXray / covid-chestxray-dataset-master / selectedimg / COVID+

Name	Last Modified	File size
..	seconds ago	

The notebook list is empty.

Fig13: folder before image selection (+ Covid images).

The folder after selecting the + Covid images



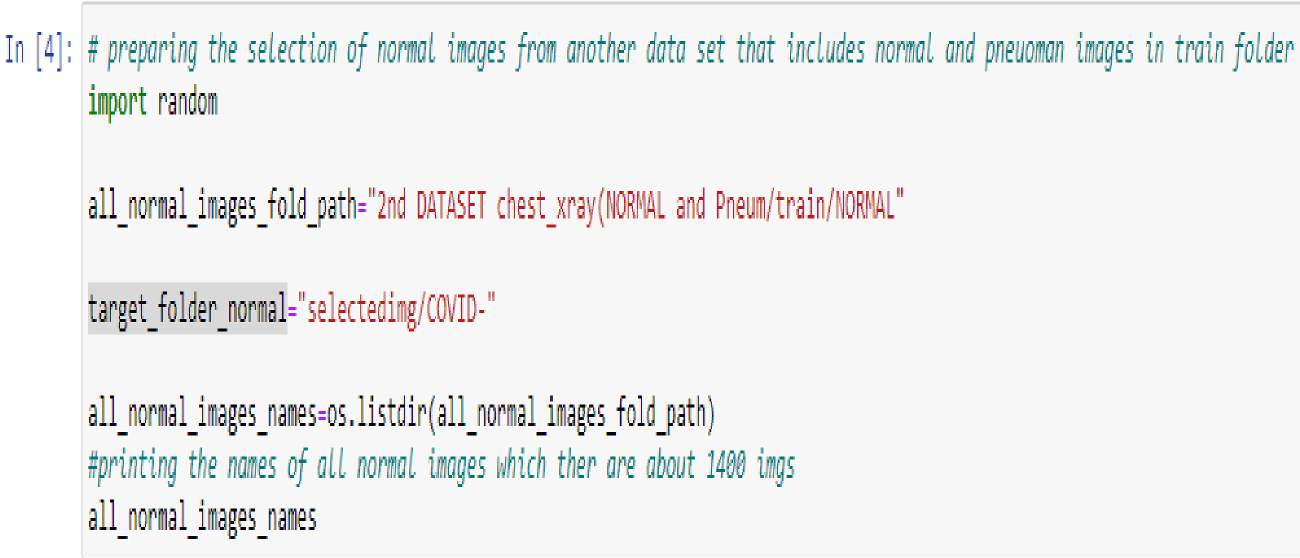
/ Desktop / graduation materials / ai-covid 19 / datasetChestXray / covid-chestxray-dataset-master / selectedimg / COVID+

	Name	Last Modified	File size
	..	seconds ago	
□	01E392EE-69F9-4E33-BFCE-E5C968654078.jpeg	2 years ago	367 kB
□	03BF7561-A9BA-4C3C-B8A0-D3E585F73F3C.jpeg	2 years ago	312 kB
□	0a7faa2a.jpg	2 years ago	729 kB
□	1-s2.0-S0140673620303706-fx1_lrg.jpg	2 years ago	813 kB
□	1-s2.0-S1684118220300608-main.pdf-001.jpg	2 years ago	13.5 kB
□	1-s2.0-S1684118220300608-main.pdf-002.jpg	2 years ago	18 kB
□	1-s2.0-S1684118220300682-main.pdf-002-a1.png	2 years ago	132 kB
□	1-s2.0-S1684118220300682-main.pdf-002-a2.png	2 years ago	148 kB
□	1-s2.0-S1684118220300682-main.pdf-003-b1.png	2 years ago	150 kB
□	1-s2.0-S1684118220300682-main.pdf-003-b2.png	2 years ago	164 kB
□	1-s2.0-S2214250920300834-gr1_lrg-b.png	2 years ago	471 kB
□	1-s2.0-S2214250920300834-gr1_lrg-c.png	2 years ago	440 kB
□	1-s2.0-S2387020620301959-gr4_lrg-b.png	2 years ago	121 kB
□	1-s2.0-S2387020620301959-gr4_lrg-c.png	2 years ago	128 kB
□	1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-000-fig1a.png	2 years ago	203 kB
□	1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-000-fig1b.png	2 years ago	207 kB

Fig14: folder after image selection (+ Covid images).

2-5 Python Code for preparing the (NORMAL) (-) covid-19 dataset:-

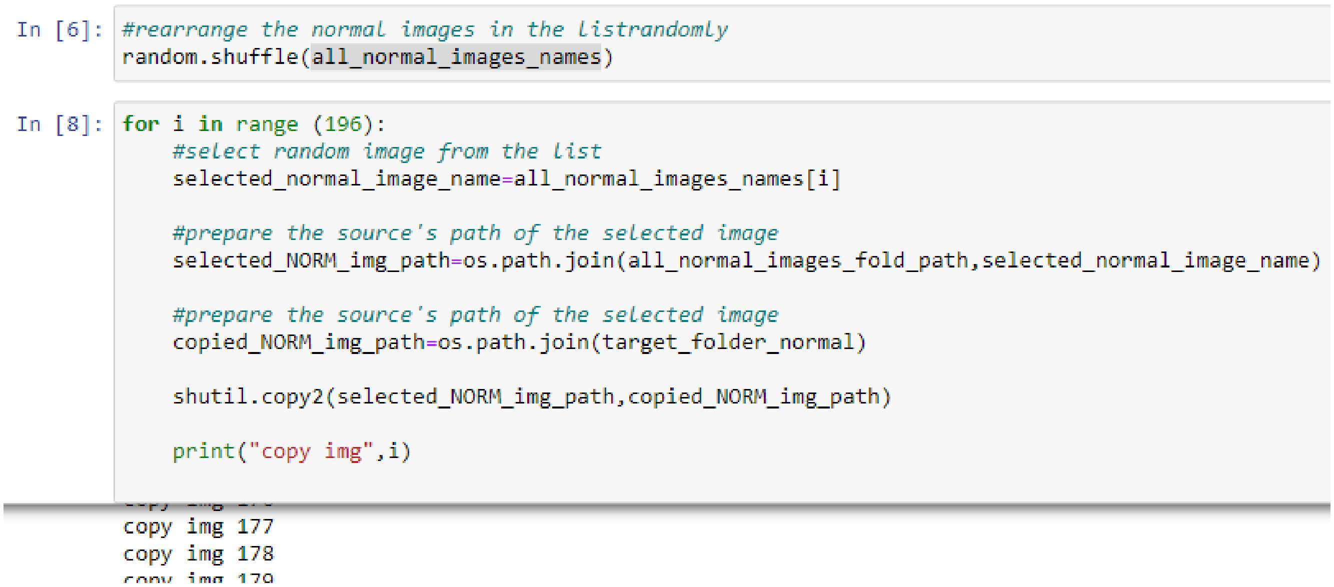
1- represent the images’ names in list



```
In [4]: # preparing the selection of normal images from another data set that includes normal and pneuoman images in train folder
        import random

        all_normal_images_fold_path="2nd DATASET chest_xray(NORMAL and Pneum/train/NORMAL"

        target_folder_normal="selectedimg/COVID-"

        all_normal_images_names=os.listdir(all_normal_images_fold_path)
        #printing the names of all normal images which ther are about 1400 imgs
        all_normal_images_names
```

Import random	Enable using random function for random selection
all_norm_img_fold_path	Holds the directory path of the folder of normal images to be sent
Target_folder_normal	Holds the directory path of the folder that will recive normal images
all_norm_imgs_names	Holds the names of all the normal images in the source folder

2- represent the images' names in list



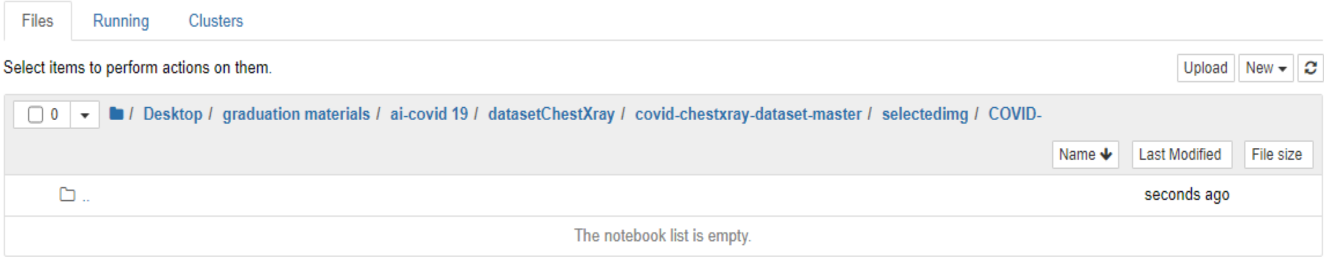
```
In [6]: #rearrange the normal images in the listrandomly
        random.shuffle(all_normal_images_names)

In [8]: for i in range (196):
            #select random image from the list
            selected_normal_image_name=all_normal_images_names[i]

            #prepare the source's path of the selected image
            selected_NORM_img_path=os.path.join(all_normal_images_fold_path,selected_normal_image_name)

            #prepare the source's path of the selected image
            copied_NORM_img_path=os.path.join(target_folder_normal)

            shutil.copy2(selected_NORM_img_path,copied_NORM_img_path)

            print("copy img",i)

        copy img 177
        copy img 178
        copy img 179
```

Random.shufle()	Function that responsible for rearrange the images randomly
Selected_norm_img_name	The variable that receives a random name of image to be copied later in our folder
Selected_norm_img_path	Prepare the directory path of the selected image in source folder
Copied_norm_img_path	Prepare the directory path of the selected image in destination folder
Shutil.copy2()	Function to copy the file from path(source) to another path(destination)

Conclusion

Among 1000 Chest-Xray of normal images:
select 196 normal Chest-Xray and copy them into another folder (-) COVID.

The folder before selecting the (normal) - Covid images.



Select items to perform actions on them.

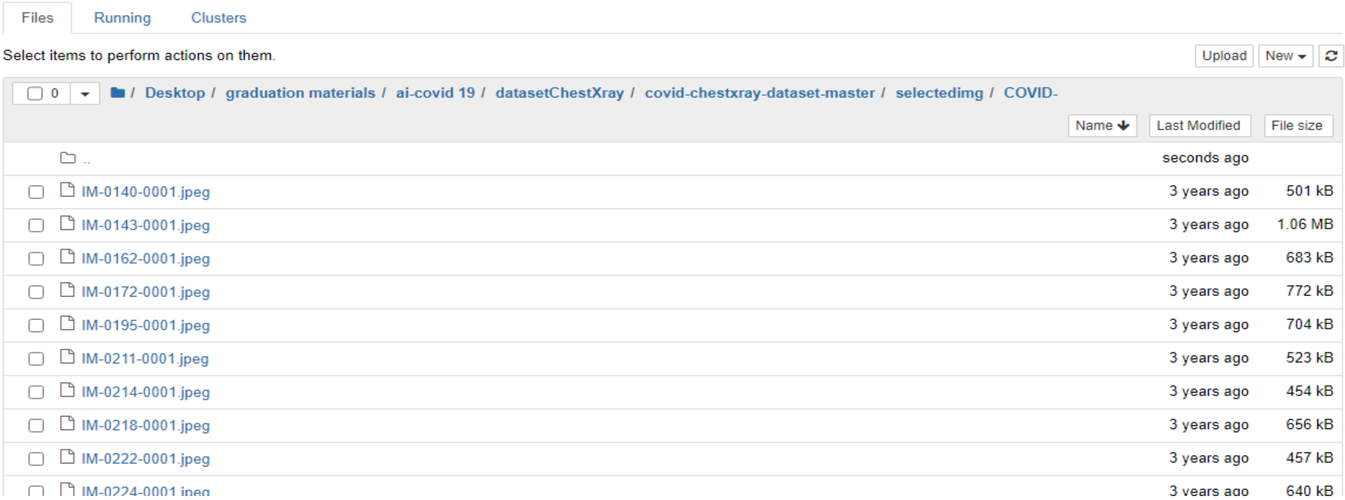
□ 0 / Desktop / graduation materials / ai-covid 19 / datasetChestXray / covid-chestxray-dataset-master / selectedimg / COVID-

Name	Last Modified	File size
..	seconds ago	

The notebook list is empty.

Fig15: folder before image selection (Normal images).

The folder before selecting the (normal) - Covid images.



Select items to perform actions on them.

□ 0 / Desktop / graduation materials / ai-covid 19 / datasetChestXray / covid-chestxray-dataset-master / selectedimg / COVID-

	Name	Last Modified	File size
	..	seconds ago	
□	IM-0140-0001.jpeg	3 years ago	501 kB
□	IM-0143-0001.jpeg	3 years ago	1.06 MB
□	IM-0162-0001.jpeg	3 years ago	683 kB
□	IM-0172-0001.jpeg	3 years ago	772 kB
□	IM-0195-0001.jpeg	3 years ago	704 kB
□	IM-0211-0001.jpeg	3 years ago	523 kB
□	IM-0214-0001.jpeg	3 years ago	454 kB
□	IM-0218-0001.jpeg	3 years ago	656 kB
□	IM-0222-0001.jpeg	3 years ago	457 kB
□	IM-0224-0001.jpeg	3 years ago	640 kB

Fig16: folder after image selection (Normal images).

3-Conclusion of Image selection:-

Now we have two folders:-

The first folder called COVID(+):-

it contains 196 chest X-rays images that are diagnoised as COVID-19.

The second folder called COVID(-):- it contains about 196 chest X-rays images that are diagnoised as Normal.

2.1.2.2-Image Splitting: -

1-introduction

The artifical intelligence model requries train data set and test dataset and based on that and After having a data set for the chest-Xrays of normal images and covid images, we will split the dataset by using spyder program.

2-Technique of Image Divation:-

2-1 Spyder IDE introduction:-

Spyder is an open-source cross-platform IDE, written completely in Python. It is designed for scientists, data analysts, and engineers.

2-2 Spyder IDE operation:-

It's used to receive a path of our dataset's folder in a certain variable then by using the libarary splitfolders.ratio(), it will divide the folder's contents into:

The first folder called Train:- it contains two folder:

COVID(+): about 0.7*196 chest X-rays images that are diagnoised as COVID-19

COVID(-): about 0.7*196 chest X-rays images that are diagnoised as NORMAL

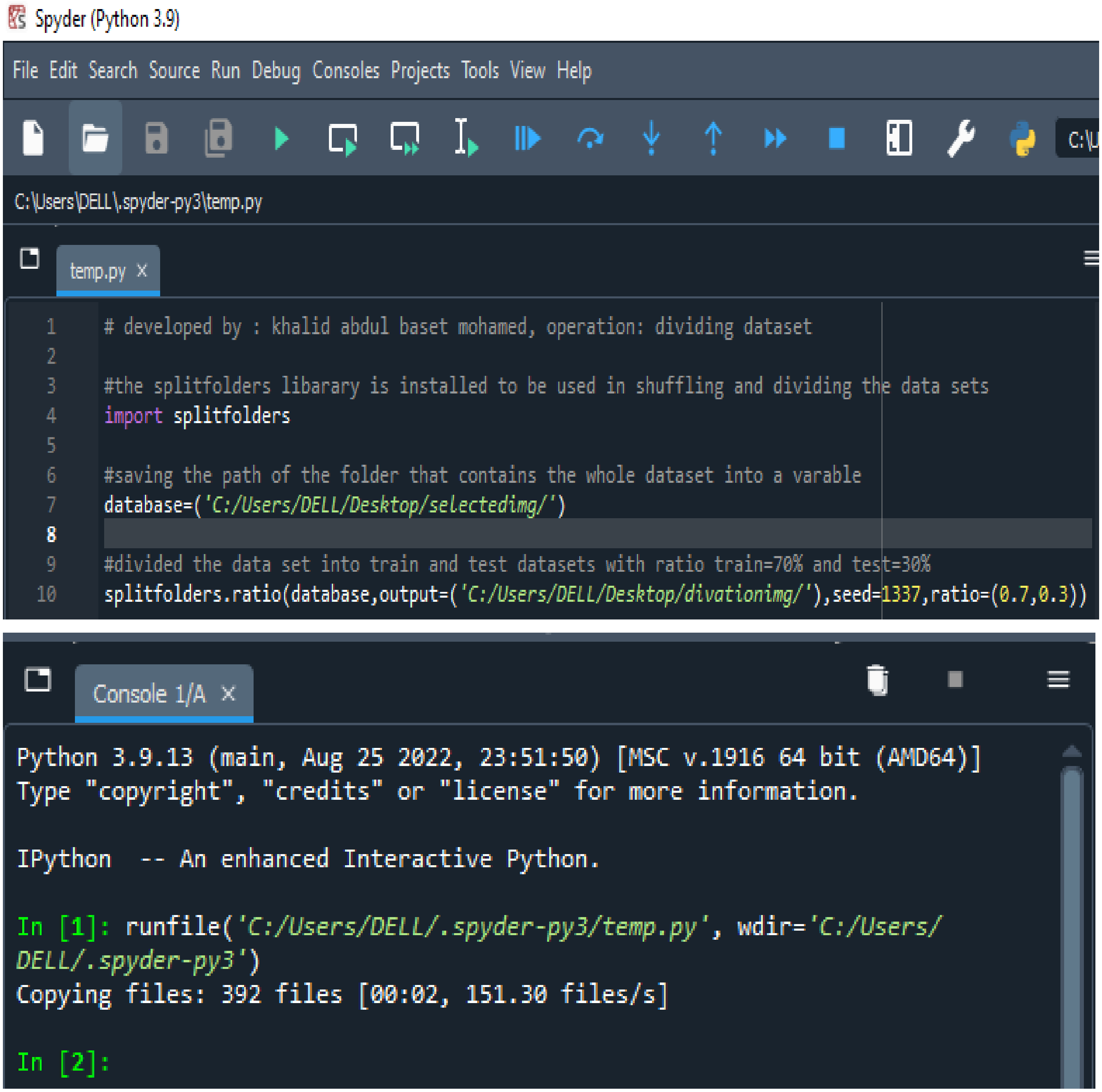
The second folder called test:- it contains two folder:

COVID(+): about 0.3*196 chest X-rays images that are diagnoised as COVID-19

COVID(-): about 0.3*196 chest X-rays images that are diagnoised as NORMAL

2-3 Spyder IDE's screen shot of the outputs:-

1-the code of divation dataset:-



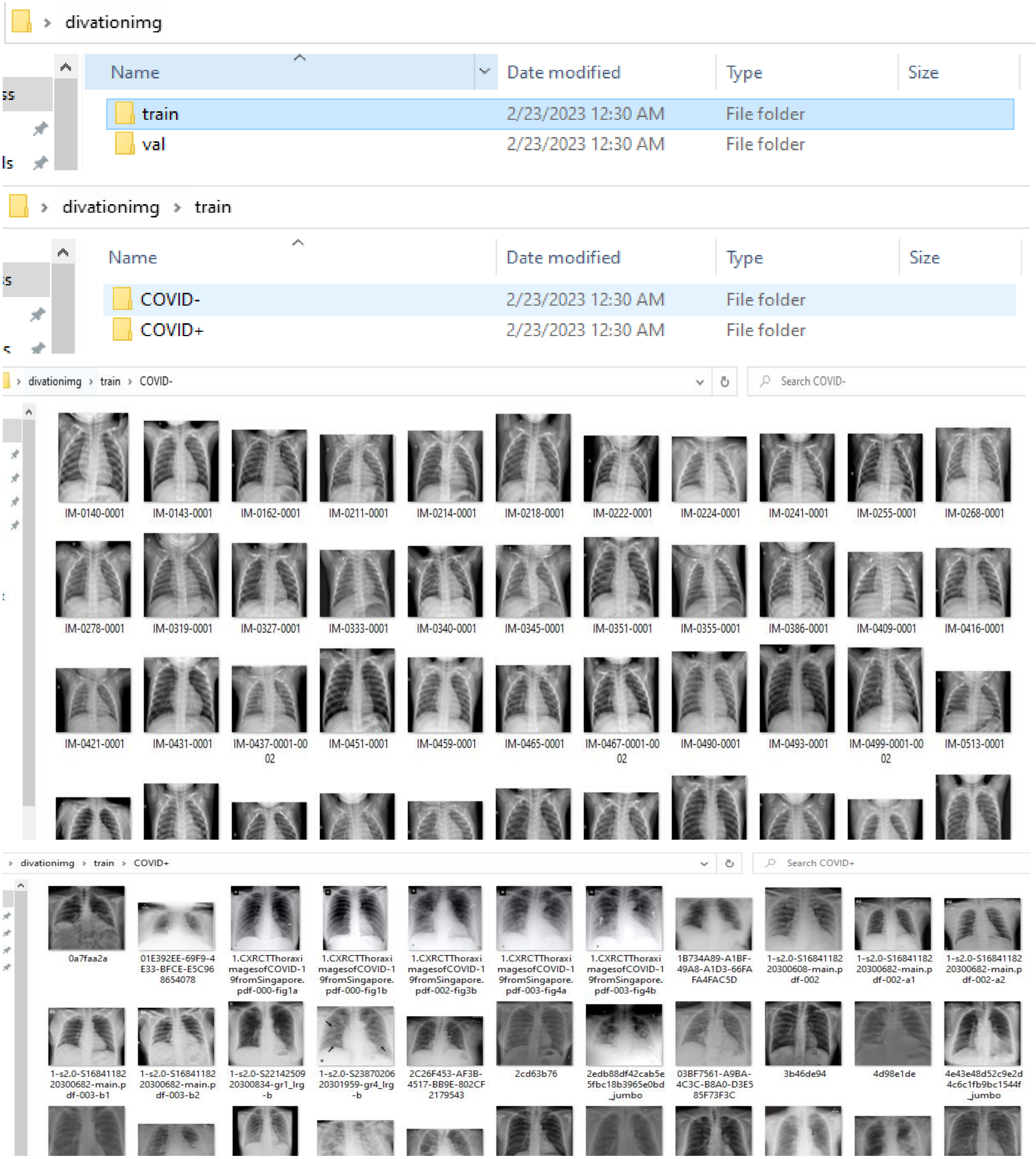
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\U
C:\Users\DELL\.spyder-py3\temp.py
temp.py
Console 1/A

```python
# developed by : khalid abdul baset mohamed, operation: dividing dataset

#the splitfolders libarary is installed to be used in shuffling and dividing the data sets
import splitfolders

#saving the path of the folder that contains the whole dataset into a varable
database=('C:/Users/DELL/Desktop/selectedimg/')

#divided the data set into train and test datasets with ratio train=70% and test=30%
splitfolders.ratio(database,output=('C:/Users/DELL/Desktop/divationimg/'),seed=1337,ratio=(0.7,0.3))
```

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython  -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/DELL/.spyder-py3/temp.py', wdir='C:/Users/
DELL/.spyder-py3')
Copying files: 392 files [00:02, 151.30 files/s]

In [2]:
```

Fig17: Operation of data splitting on spyder program.

2-the result of divation dataset:-



divationimg

Name	Date modified	Type	Size
train	2/23/2023 12:30 AM	File folder	
val	2/23/2023 12:30 AM	File folder	

divationimg › train

Name	Date modified	Type	Size
COVID-	2/23/2023 12:30 AM	File folder	
COVID+	2/23/2023 12:30 AM	File folder	

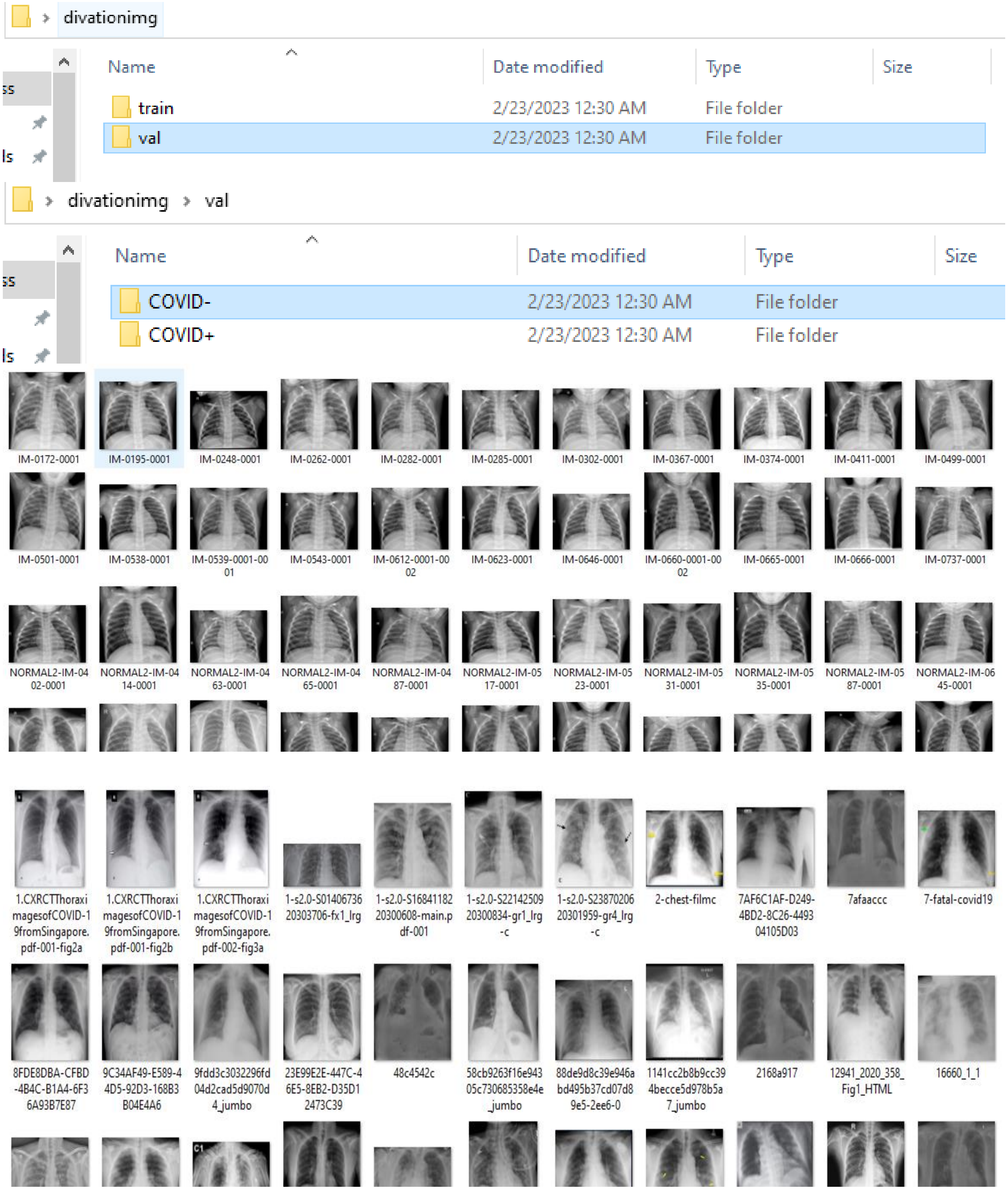
divationimg › train › COVID-

IM-0140-0001
IM-0143-0001
IM-0162-0001
IM-0211-0001
IM-0214-0001
IM-0218-0001
IM-0222-0001
IM-0224-0001
IM-0241-0001
IM-0255-0001
IM-0268-0001
IM-0278-0001
IM-0319-0001
IM-0327-0001
IM-0333-0001
IM-0340-0001
IM-0345-0001
IM-0351-0001
IM-0355-0001
IM-0386-0001
IM-0409-0001
IM-0416-0001
IM-0421-0001
IM-0431-0001
IM-0437-0001-0002
IM-0451-0001
IM-0459-0001
IM-0465-0001
IM-0467-0001-0002
IM-0490-0001
IM-0493-0001
IM-0499-0001-0002
IM-0513-0001

divationimg › train › COVID+

0a7faa2a
01E392EE-69F9-4E33-BFCE-E5C968654078
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-000-fig1a
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-000-fig1b
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-002-fig3b
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-003-fig4a
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-003-fig4b
1B734A89-A1BF-49A8-A1D3-66FAFA4FAC5D
1-s2.0-S1684118220300608-main.pdf-002
1-s2.0-S1684118220300682-main.pdf-002-a1
1-s2.0-S1684118220300682-main.pdf-002-a2
1-s2.0-S1684118220300682-main.pdf-003-b1
1-s2.0-S1684118220300682-main.pdf-003-b2
1-s2.0-S2214250920300834-gr1_lrg-b
1-s2.0-S2387020620301959-gr4_lrg-b
2C26F453-AF3B-4517-BB9E-802CF2179543
2cd63b76
2edb88df42cab5e5fbc18b3965e0bd_jumbo
03BF7561-A9BA-4C3C-B8A0-D3E585F73F3C
3b46de94
4d98e1de
4e43e48d52c9e2d4c6c1fb9bc1544f_jumbo

Fig18: output of data splitting on spyder program (train folder).



divationimg

Name	Date modified	Type	Size
train	2/23/2023 12:30 AM	File folder	
val	2/23/2023 12:30 AM	File folder	

divationimg › val

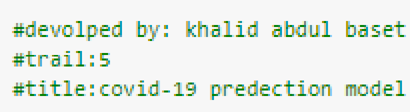
Name	Date modified	Type	Size
COVID-	2/23/2023 12:30 AM	File folder	
COVID+	2/23/2023 12:30 AM	File folder	

IM-0172-0001
IM-0195-0001
IM-0248-0001
IM-0262-0001
IM-0282-0001
IM-0285-0001
IM-0302-0001
IM-0367-0001
IM-0374-0001
IM-0411-0001
IM-0499-0001
IM-0501-0001
IM-0538-0001
IM-0539-0001-0001
IM-0543-0001
IM-0612-0001-0002
IM-0623-0001
IM-0646-0001
IM-0660-0001-0002
IM-0665-0001
IM-0666-0001
IM-0737-0001
NORMAL2-IM-0402-0001
NORMAL2-IM-0414-0001
NORMAL2-IM-0463-0001
NORMAL2-IM-0465-0001
NORMAL2-IM-0487-0001
NORMAL2-IM-0517-0001
NORMAL2-IM-0523-0001
NORMAL2-IM-0531-0001
NORMAL2-IM-0535-0001
NORMAL2-IM-0587-0001
NORMAL2-IM-0645-0001
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-001-fig2a
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-001-fig2b
1.CXRCTThoraximagesofCOVID-19fromSingapore.pdf-002-fig3a
1-s2.0-S0140673620303706-fx1_lrg
1-s2.0-S1684118220300608-main.pdf-001
1-s2.0-S2214250920300834-gr1_lrg-c
1-s2.0-S2387020620301959-gr4_lrg-c
2-chest-filmc
7AF6C1AF-D249-4BD2-8C26-449304105D03
7afaaccc
7-fatal-covid19
8FDE8DBA-CFBD-4B4C-B1A4-6F36A93B7E87
9C34AF49-E589-44D5-92D3-168B3B04E4A6
9fdd3c3032296fd04d2cad5d9070d4_jumbo
23E99E2E-447C-46E5-8EB2-D35D12473C39
48c4542c
58cb9263f16e94305c730685358e4e_jumbo
88de9d8c39e946abd495b37cd07d89e5-2ee6-0
1141cc2b8b9cc394becce5d978b5a7_jumbo
2168a917
12941_2020_358_Fig1_HTML
16660_1_1
C1

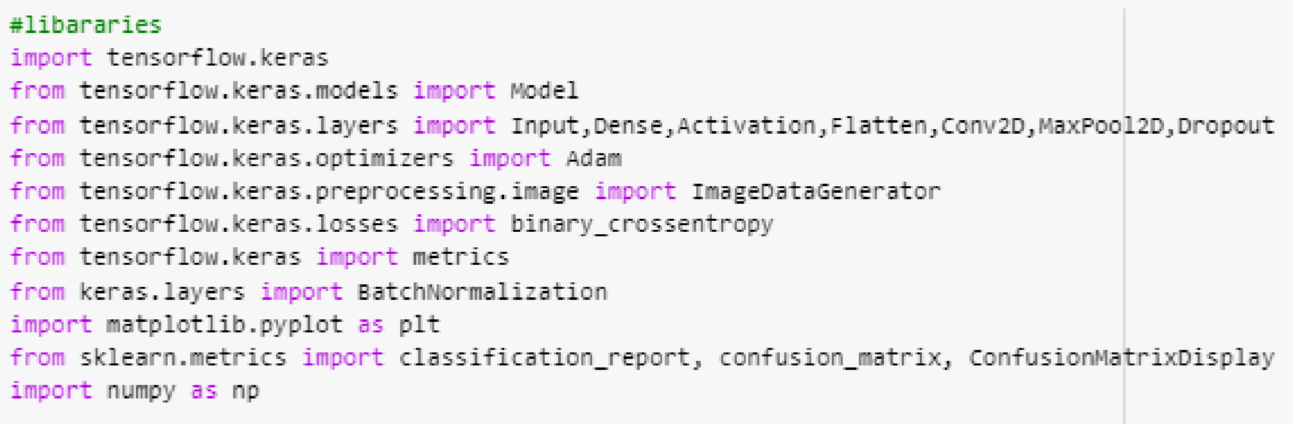
Fig19: output of data splitting on spyder program (train folder).

2.1.3- Machine Learning Model: -

2.1.3.1 – Import Libraries: -

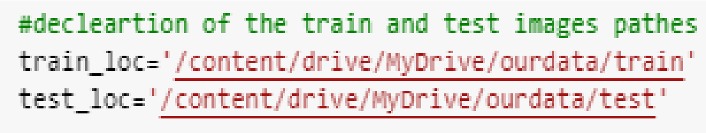


```
#devolped by: khalid abdul baset
#trail:5
#title:covid-19 predection model
```



```
#libararies
import tensorflow.keras
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input,Dense,Activation,Flatten,Conv2D,MaxPool2D,Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.losses import binary_crossentropy
from tensorflow.keras import metrics
from keras.layers import BatchNormalization
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
import numpy as np
```

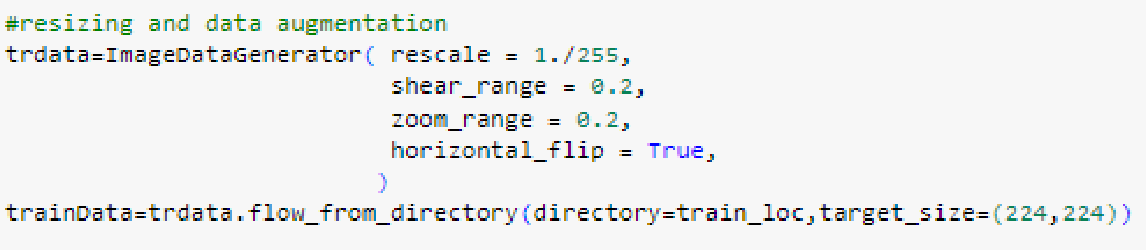
2.1.3.2 – Load training and test Data: -



```
#declеartion of the train and test images pathes
train_loc='/content/drive/MyDrive/ourdata/train'
test_loc='/content/drive/MyDrive/ourdata/test'
```

2.1.3.3 – Resizing the images and data augmentation: -

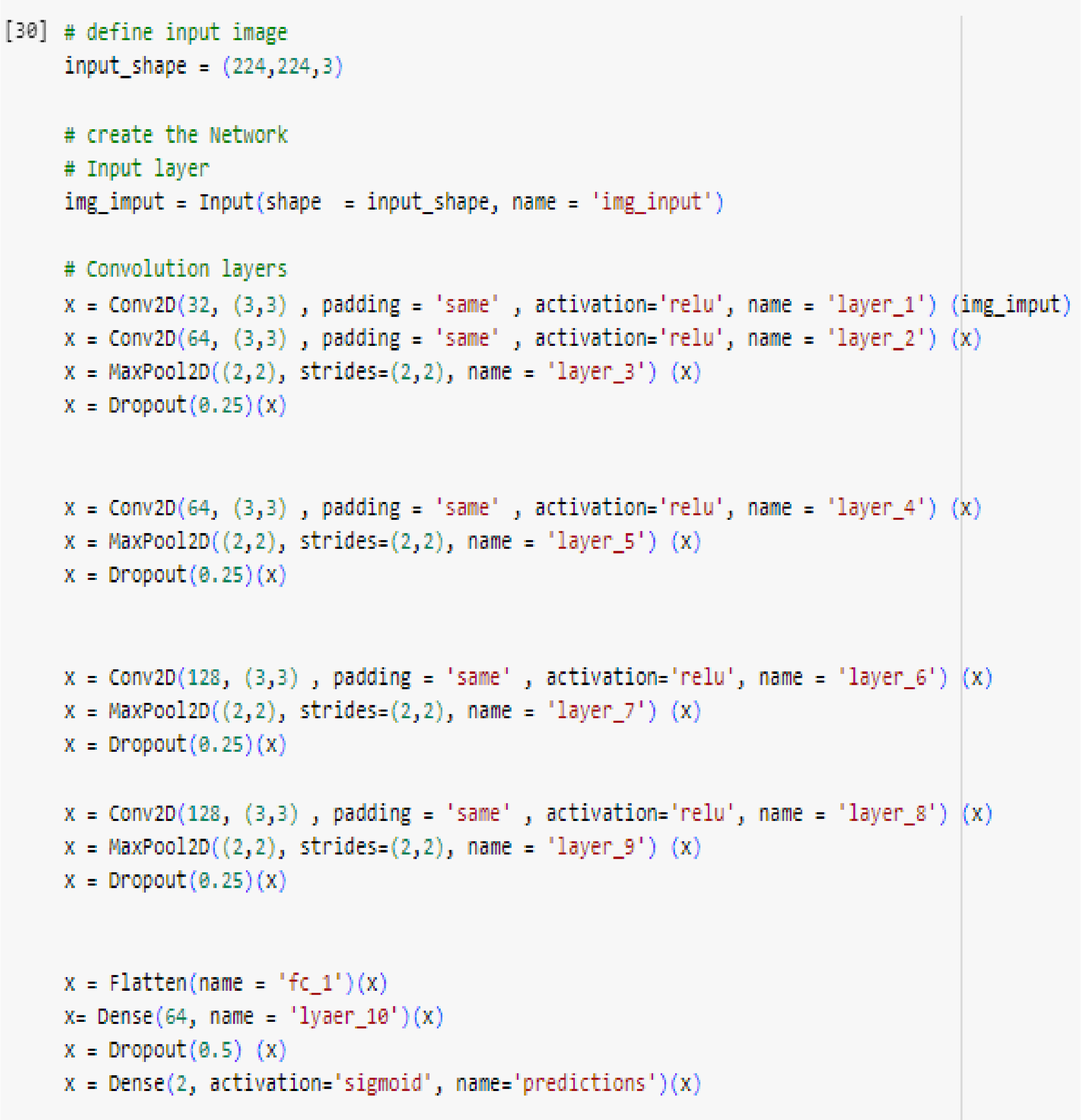
Data augmentation:
it is the process of expanding dataset to improve the model skills in generalization.



```
#resizing and data augmentation
trdata=ImageDataGenerator( rescale = 1./255,
                           shear_range = 0.2,
                           zoom_range = 0.2,
                           horizontal_flip = True,
                          )
trainData=trdata.flow_from_directory(directory=train_loc,target_size=(224,224))
```

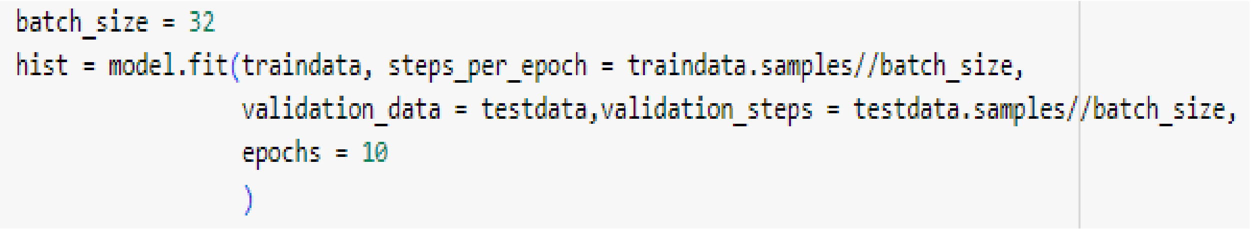
-

2.1.3.4 – Model Architecture: -

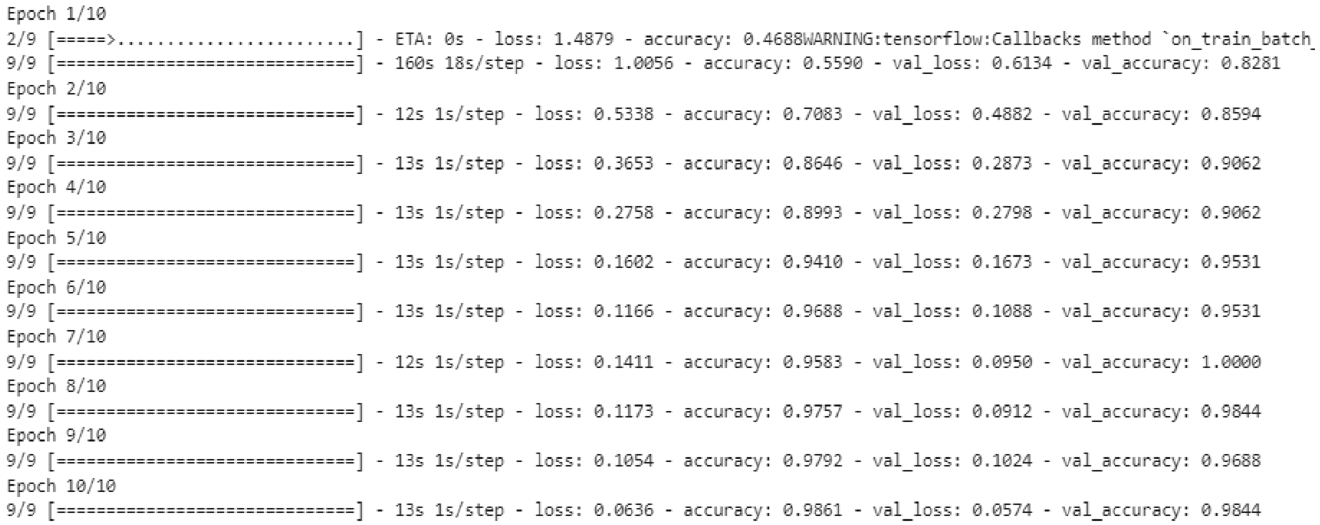


```
[30] # define input image
     input_shape = (224,224,3)

     # create the Network
     # Input layer
     img_imput = Input(shape  = input_shape, name = 'img_input')

     # Convolution layers
     x = Conv2D(32, (3,3) , padding = 'same' , activation='relu', name = 'layer_1') (img_imput)
     x = Conv2D(64, (3,3) , padding = 'same' , activation='relu', name = 'layer_2') (x)
     x = MaxPool2D((2,2), strides=(2,2), name = 'layer_3') (x)
     x = Dropout(0.25)(x)


     x = Conv2D(64, (3,3) , padding = 'same' , activation='relu', name = 'layer_4') (x)
     x = MaxPool2D((2,2), strides=(2,2), name = 'layer_5') (x)
     x = Dropout(0.25)(x)


     x = Conv2D(128, (3,3) , padding = 'same' , activation='relu', name = 'layer_6') (x)
     x = MaxPool2D((2,2), strides=(2,2), name = 'layer_7') (x)
     x = Dropout(0.25)(x)

     x = Conv2D(128, (3,3) , padding = 'same' , activation='relu', name = 'layer_8') (x)
     x = MaxPool2D((2,2), strides=(2,2), name = 'layer_9') (x)
     x = Dropout(0.25)(x)


     x = Flatten(name = 'fc_1')(x)
     x= Dense(64, name = 'lyaer_10')(x)
     x = Dropout(0.5) (x)
     x = Dense(2, activation='sigmoid', name='predictions')(x)
```

2.1.3.8 – Model Fitting: -

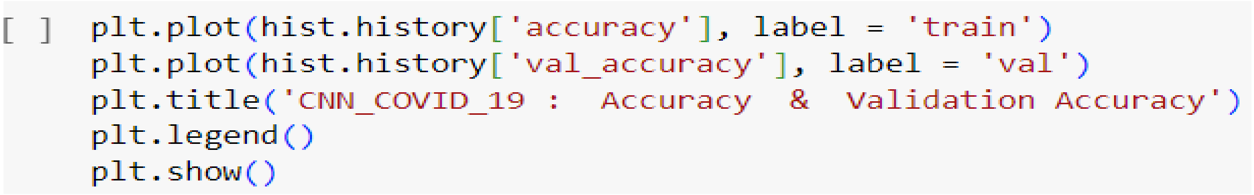


```
batch_size = 32
hist = model.fit(traindata, steps_per_epoch = traindata.samples//batch_size,
                 validation_data = testdata,validation_steps = testdata.samples//batch_size,
                 epochs = 10
                 )
```

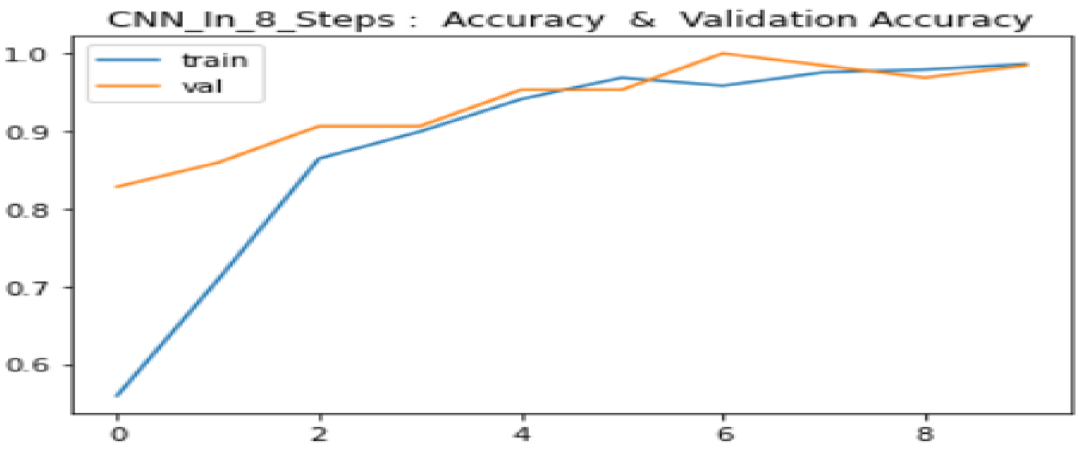


```
Epoch 1/10
2/9 [=====>........................] - ETA: 0s - loss: 1.4879 - accuracy: 0.4688WARNING:tensorflow:Callbacks method `on_train_batch_
9/9 [==============================] - 160s 18s/step - loss: 1.0056 - accuracy: 0.5590 - val_loss: 0.6134 - val_accuracy: 0.8281
Epoch 2/10
9/9 [==============================] - 12s 1s/step - loss: 0.5338 - accuracy: 0.7083 - val_loss: 0.4882 - val_accuracy: 0.8594
Epoch 3/10
9/9 [==============================] - 13s 1s/step - loss: 0.3653 - accuracy: 0.8646 - val_loss: 0.2873 - val_accuracy: 0.9062
Epoch 4/10
9/9 [==============================] - 13s 1s/step - loss: 0.2758 - accuracy: 0.8993 - val_loss: 0.2798 - val_accuracy: 0.9062
Epoch 5/10
9/9 [==============================] - 13s 1s/step - loss: 0.1602 - accuracy: 0.9410 - val_loss: 0.1673 - val_accuracy: 0.9531
Epoch 6/10
9/9 [==============================] - 13s 1s/step - loss: 0.1166 - accuracy: 0.9688 - val_loss: 0.1088 - val_accuracy: 0.9531
Epoch 7/10
9/9 [==============================] - 12s 1s/step - loss: 0.1411 - accuracy: 0.9583 - val_loss: 0.0950 - val_accuracy: 1.0000
Epoch 8/10
9/9 [==============================] - 13s 1s/step - loss: 0.1173 - accuracy: 0.9757 - val_loss: 0.0912 - val_accuracy: 0.9844
Epoch 9/10
9/9 [==============================] - 13s 1s/step - loss: 0.1054 - accuracy: 0.9792 - val_loss: 0.1024 - val_accuracy: 0.9688
Epoch 10/10
9/9 [==============================] - 13s 1s/step - loss: 0.0636 - accuracy: 0.9861 - val_loss: 0.0574 - val_accuracy: 0.9844
```

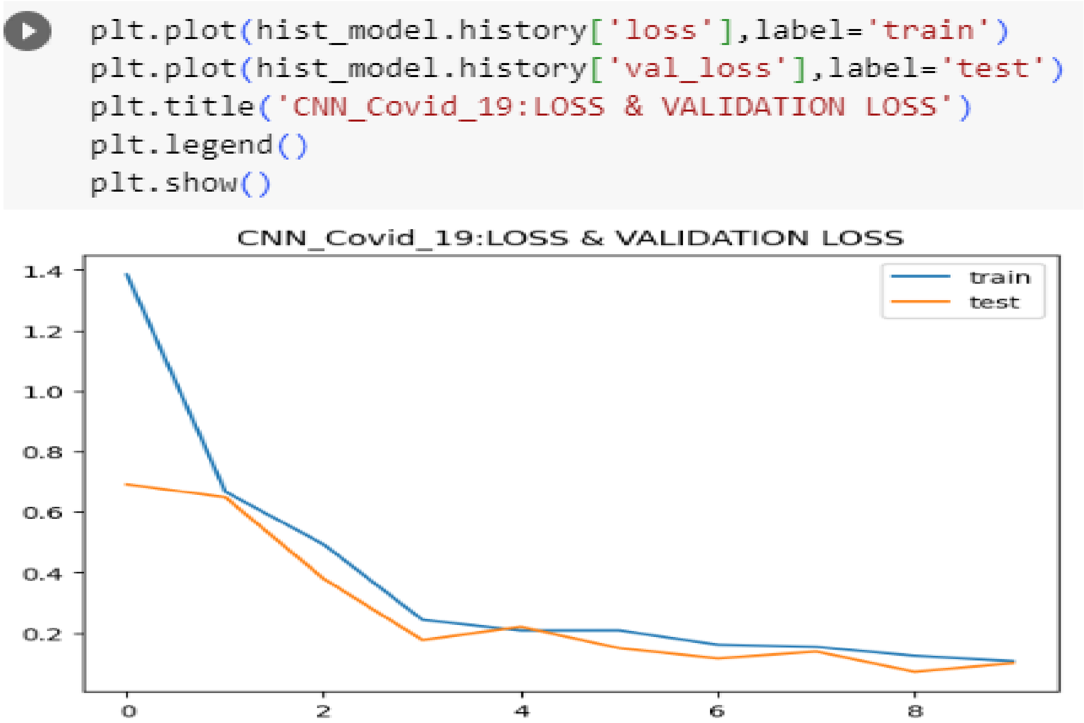
2.1.3.8 – Model Accuracy: -



```
[ ]  plt.plot(hist.history['accuracy'], label = 'train')
     plt.plot(hist.history['val_accuracy'], label = 'val')
     plt.title('CNN_COVID_19 :  Accuracy  &  Validation Accuracy')
     plt.legend()
     plt.show()
```

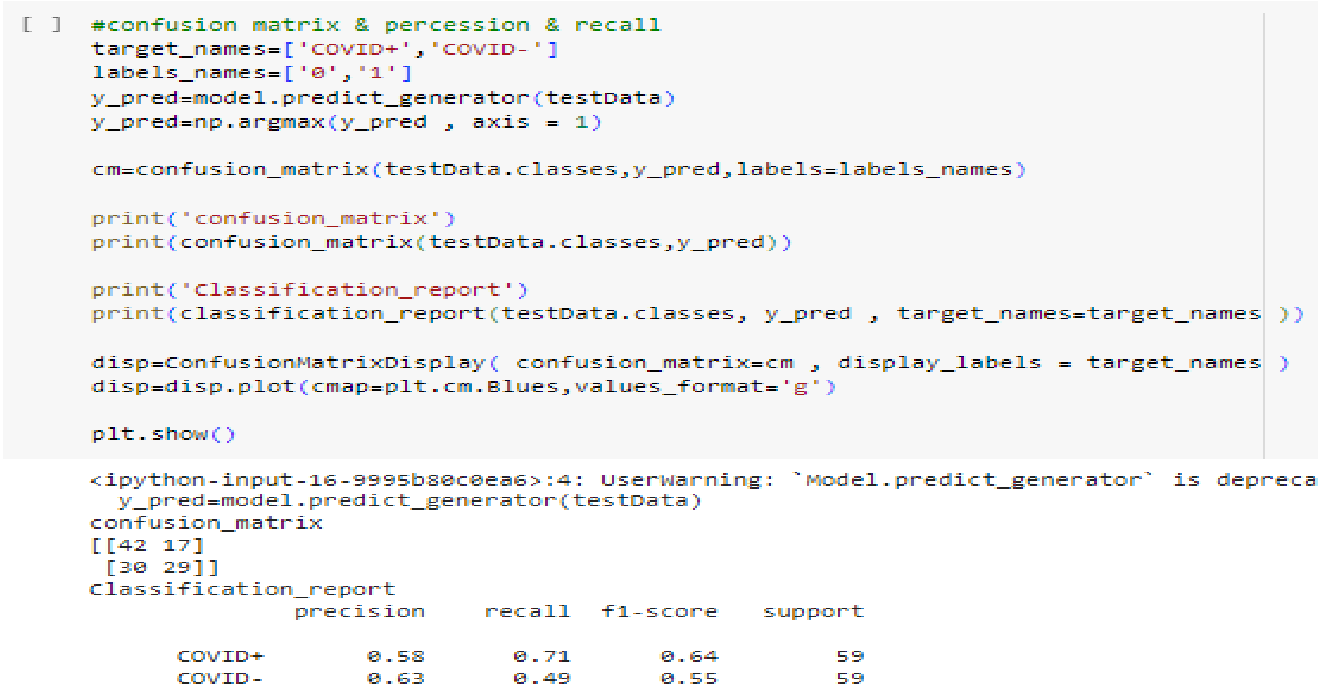


2.1.3.9 – Model Loss: -



```
plt.plot(hist_model.history['loss'],label='train')
plt.plot(hist_model.history['val_loss'],label='test')
plt.title('CNN_Covid_19:LOSS & VALIDATION LOSS')
plt.legend()
plt.show()
```

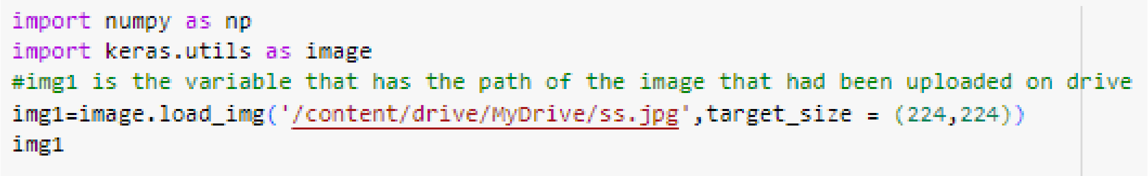
2.1.3.10 – Confusion matrix: -



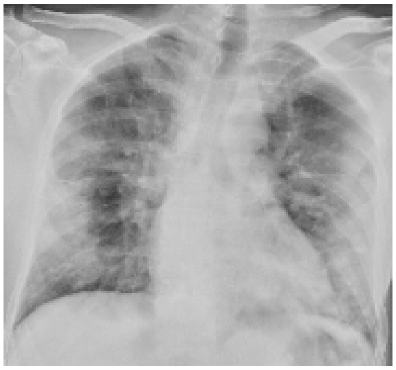
```
[ ]  #confusion matrix & percession & recall
     target_names=['COVID+','COVID-']
     labels_names=['0','1']
     y_pred=model.predict_generator(testData)
     y_pred=np.argmax(y_pred , axis = 1)

     cm=confusion_matrix(testData.classes,y_pred,labels=labels_names)

     print('confusion_matrix')
     print(confusion_matrix(testData.classes,y_pred))

     print('Classification_report')
     print(classification_report(testData.classes, y_pred , target_names=target_names ))

     disp=ConfusionMatrixDisplay( confusion_matrix=cm , display_labels = target_names )
     disp=disp.plot(cmap=plt.cm.Blues,values_format='g')

     plt.show()
```

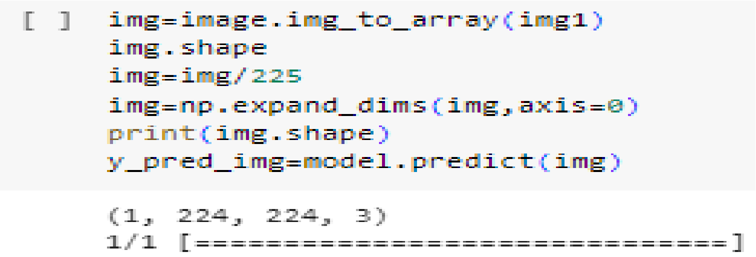
```
<ipython-input-16-9995b80c0ea6>:4: UserWarning: `Model.predict_generator` is depreca
  y_pred=model.predict_generator(testData)
confusion_matrix
[[42 17]
 [30 29]]
Classification_report
              precision    recall  f1-score   support

      COVID+       0.58      0.71      0.64        59
      COVID-       0.63      0.49      0.55        59
```

2.1.3.11 – Testing the model: -

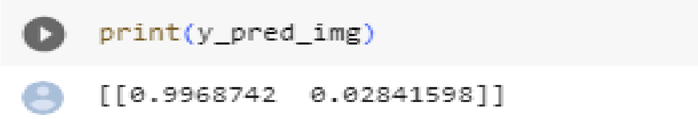


```
import numpy as np
import keras.utils as image
#img1 is the variable that has the path of the image that had been uploaded on drive
img1=image.load_img('/content/drive/MyDrive/ss.jpg',target_size = (224,224))
img1
```

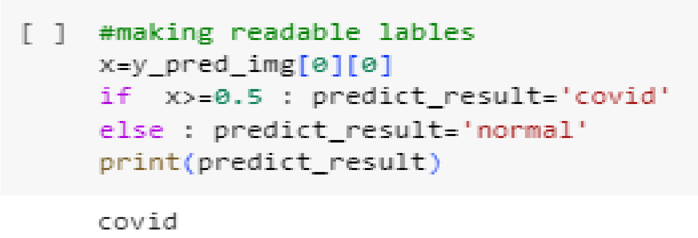




```
[ ]  img=image.img_to_array(img1)
     img.shape
     img=img/225
     img=np.expand_dims(img,axis=0)
     print(img.shape)
     y_pred_img=model.predict(img)

     (1, 224, 224, 3)
     1/1 [==============================]
```



```
print(y_pred_img)

[[0.9968742  0.02841598]]
```



```
[ ]  #making readable lables
     x=y_pred_img[0][0]
     if  x>=0.5 : predict_result='covid'
     else : predict_result='normal'
     print(predict_result)

     covid
```

5- Implementation

Chapter Two: Medicine Recommendation system

1- Introduction: -

In this application, we are going to use the natural language processing techniques to find the best matches (alternatives) medicines of whatever medicine that user seeks for, and provide him these matches with links to a certain pharmacy website so that he will not only know the alternatives of his medicines but also, he can order them as fast as possible.

The application represented as a web page that contains a search bar that user can enter his medicine and press the button recommend medicines, then the system start to recommend the closest 5th alternatives.

1.1- Problem Statement: -

It's difficult to recommend an alternative medicines for a certain medicine that a patient asks for, in case of unavailability of last one.

A plenty of medicines are produced and available in the market because of pharmacy improvement, these medicines represent a large set that make it impossible for any expert or nonexpert person to be familiar with all information of these medicines.

1.2- Objectives: -

We aim to create an application that helps pharmacist expert and ordinary patient to know the best alternatives for a certain medicine.

How?

1- Create a system that has all information about the medicines

(their: usages, mechanism of interactions, ingredients, APIs,…..etc.)

2- Make the system able to comparison between medicines information

(Using natural language processing)

3- Display the closest 5th matches for the entered medicine on a webpage

4- Enable the user to order any alternatives from a certain pharmacy.

1.3- Motivations: -

This application is dedicated to serve medical field.

1- Helps patient to know and order medicines without needing an expert.

(In case of **absence** of described medicine)

2- helps patient to get best alternative medicine suits his budget

(In case of **expensive** described medicine)

3- helps pharmacist to give the client the best alternative medicines.

(In case of **limited knowledge of** medicines)

1.4- Development Operation: -

1- machine leering

2- Back-End Deployment

3- Front-End

2- Implementations: -

2.1- Machine Learning

2.1.1- Data Collection: -

2.1.1.1-Dataset Description:

Our dataset is a large collection of medicines which contains important information that helps us in the classification and comparison between medicines to get the best matches for one medicine.

This information are gathered by the help of :

1- Egyptian medicine guide: -

- It provides us the (names, reason of use, prices, APIs) of available medicines in the Egyptian market.

1- Medical references platforms: -

- It provides mechanism of interactions of the medicine’s ingredient.
- It provides usages and dosages of the medicines.
- Examples:

  1- Drugs.com - provides user ratings and reviews for prescription and over-the-counter medications.

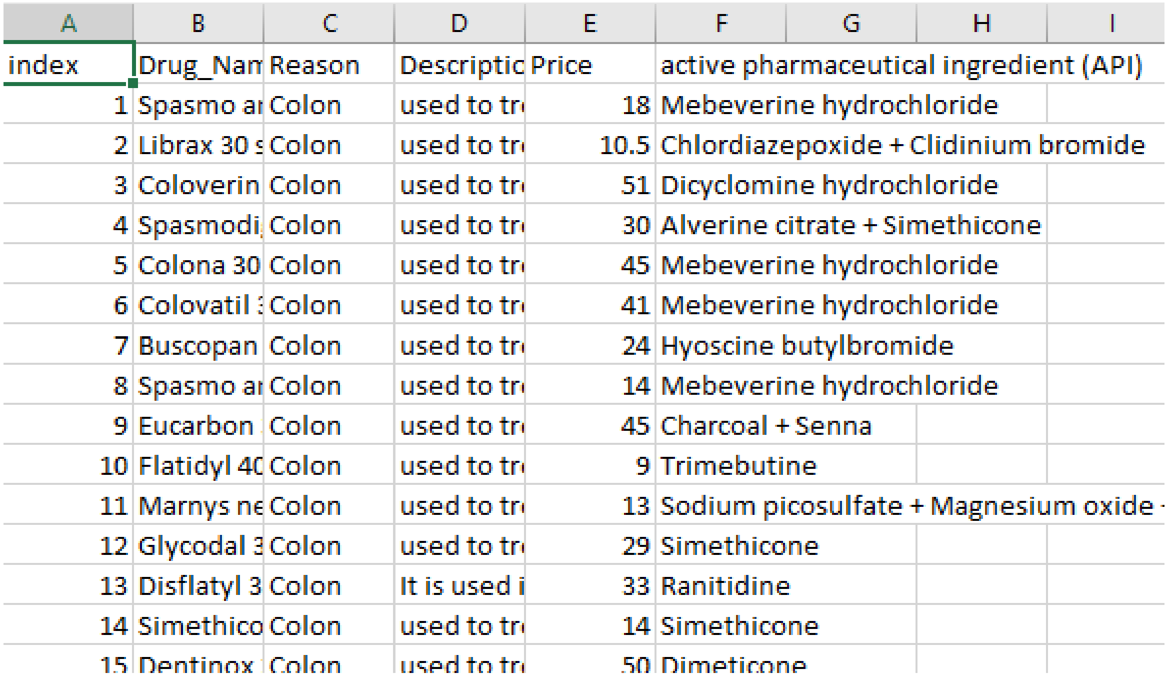
  2-WebMD - providing information about various medications, features user reviews and ratings.

  3- RxList

2.1.1.2-Dataset specifications:

1-The dataset consists of information of each medicine

(Name of drug, Reason, Description, price, APIs)



A	B	C	D	E	F	G	H	I
index	Drug_Nam	Reason	Descriptic	Price	active pharmaceutical ingredient (API)			
1	Spasmo ar	Colon	used to tr	18	Mebeverine hydrochloride			
2	Librax 30 s	Colon	used to tr	10.5	Chlordiazepoxide + Clidinium bromide			
3	Coloverin	Colon	used to tr	51	Dicyclomine hydrochloride			
4	Spasmodi	Colon	used to tr	30	Alverine citrate + Simethicone			
5	Colona 30	Colon	used to tr	45	Mebeverine hydrochloride			
6	Colovatil :	Colon	used to tr	41	Mebeverine hydrochloride			
7	Buscopan	Colon	used to tr	24	Hyoscine butylbromide			
8	Spasmo ar	Colon	used to tr	14	Mebeverine hydrochloride			
9	Eucarbon	Colon	used to tr	45	Charcoal + Senna			
10	Flatidyl 40	Colon	used to tr	9	Trimebutine			
11	Marnys ne	Colon	used to tr	13	Sodium picosulfate + Magnesium oxide -			
12	Glycodal 3	Colon	used to tr	29	Simethicone			
13	Disflatyl 3	Colon	It is used i	33	Ranitidine			
14	Simethico	Colon	used to tr	14	Simethicone			
15	Dentinox	Colon	used to tr	50	Dimeticone			

Fig20: Screen shot for the contents of medicines data set.

2-The dataset consists of about 1500 medicines

3- These medicines belong to 24 diseases categories.

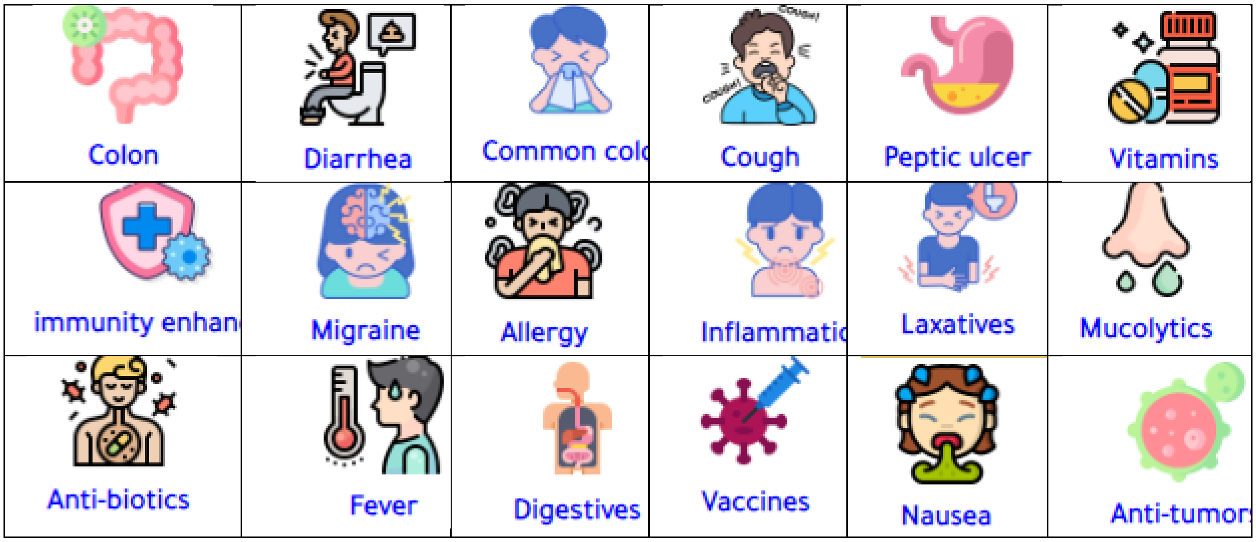
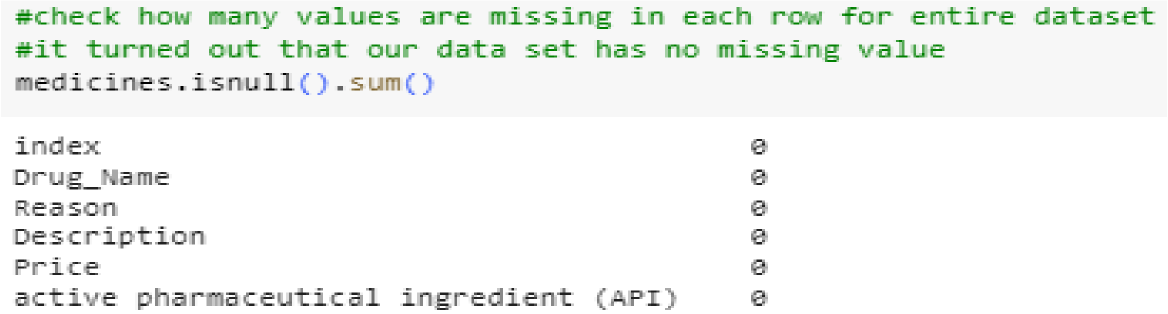


Fig21: Screen shot for the diseases category of medicines recommender.

2.1.2- Data Preprocessing (Cleansing): -

2.1.2.1-Processing of lost data: -

Check how many samples contain lost data.

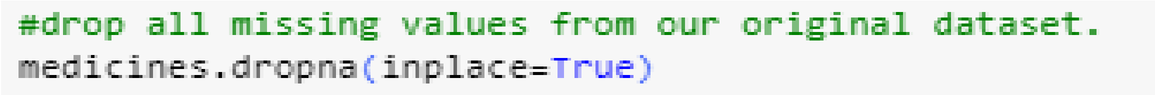


```
#check how many values are missing in each row for entire dataset
#it turned out that our data set has no missing value
medicines.isnull().sum()
```

```
index                                    0
Drug_Name                                0
Reason                                   0
Description                              0
Price                                    0
active pharmaceutical ingredient (API)   0
```

2.1.2.2-Data Filtering: -

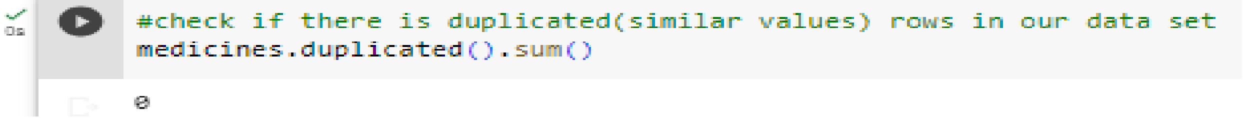
Filtering the corrupted data by dropping the samples with missing data



```
#drop all missing values from our original dataset.
medicines.dropna(inplace=True)
```

2.1.2.3-Remove duplication of Data: -

Removing the duplicated samples of medicines from the dataset to increase the efficiency of the dataset for training the model



```
#check if there is duplicated(similar values) rows in our data set
medicines.duplicated().sum()
```

```
0
```

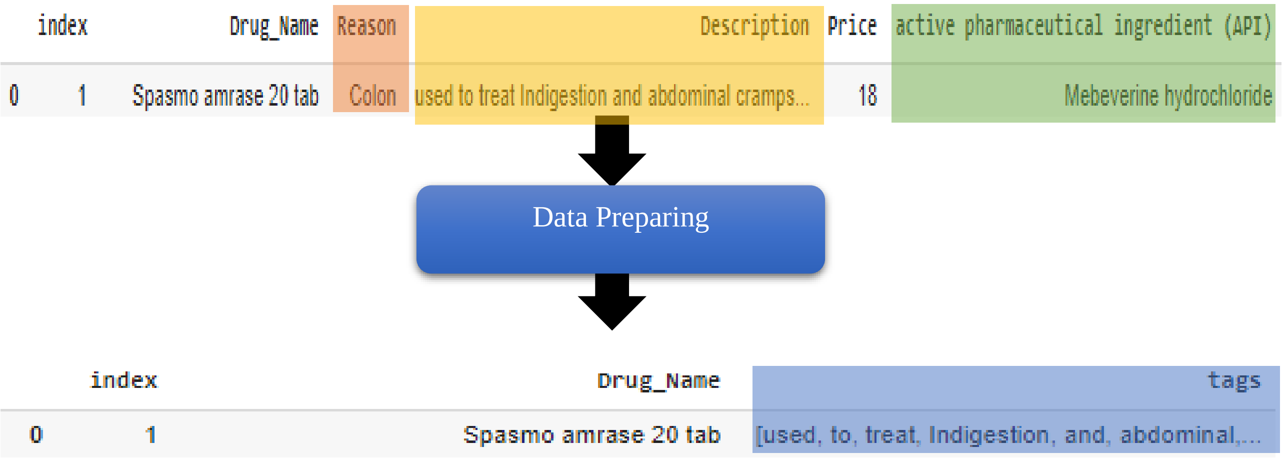
2.1.3- Data Preparing: -

2.1.3.1-Defination: -

This is the process of uniting different types of information about each medicine under variable named (**tag**).

Tag: it is the variable that contains all necessary information of medicine so the model can use to compare among different medicines.

2.1.3.2-Diagram: -



	index	Drug_Name	Reason	Description	Price	active pharmaceutical ingredient (API)
0	1	Spasmo amrase 20 tab	Colon	used to treat Indigestion and abdominal cramps...	18	Mebeverine hydrochloride

	index	Drug_Name	tags
0	1	Spasmo amrase 20 tab	[used, to, treat, Indigestion, and, abdominal,...

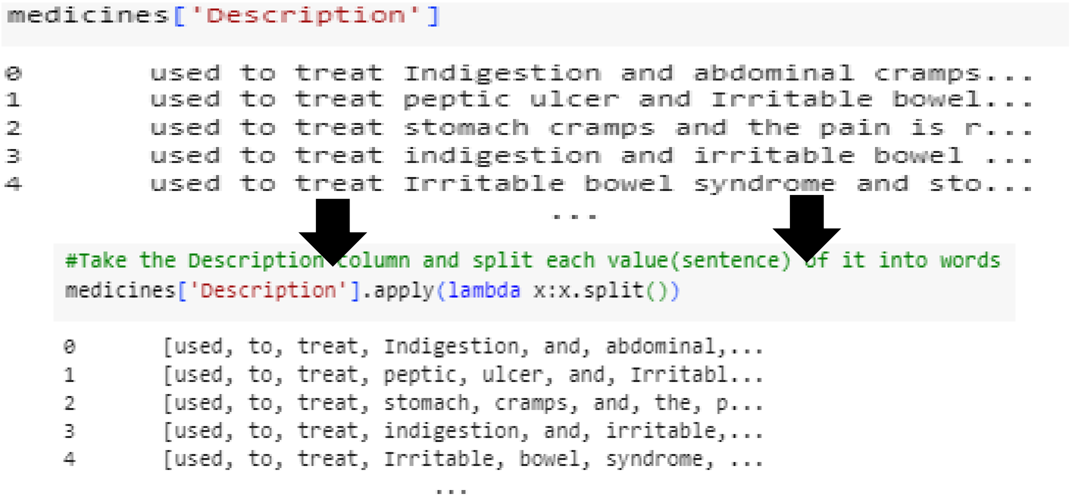
Fig22: Data preparing of medicines recommender.

2.1.3.3 - Operation: -

How to concatenate [Reason, Description, API] columns' value to represent a Tag of comparison?

1- Conversion strings of [Reason, Description, API] into list of words

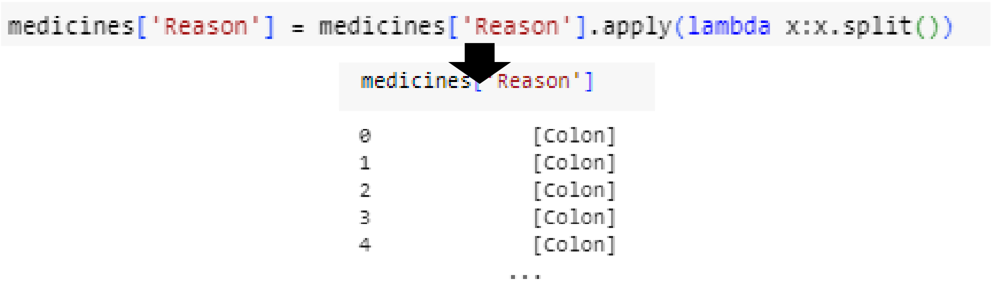
[Description column] conversion



```
medicines['Description']

0        used to treat Indigestion and abdominal cramps...
1        used to treat peptic ulcer and Irritable bowel...
2        used to treat stomach cramps and the pain is r...
3        used to treat indigestion and irritable bowel ...
4        used to treat Irritable bowel syndrome and sto...
                                 ...
```

```
#Take the Description column and split each value(sentence) of it into words
medicines['Description'].apply(lambda x:x.split())

0       [used, to, treat, Indigestion, and, abdominal,...
1       [used, to, treat, peptic, ulcer, and, Irritabl...
2       [used, to, treat, stomach, cramps, and, the, p...
3       [used, to, treat, indigestion, and, irritable,...
4       [used, to, treat, Irritable, bowel, syndrome, ...
                            ...
```

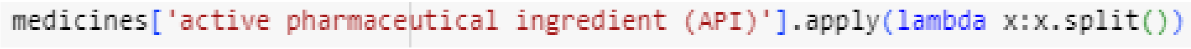
[Reason column] conversion



```
medicines['Reason'] = medicines['Reason'].apply(lambda x:x.split())
```

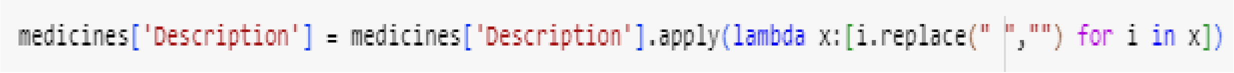
```
medicines['Reason']

0             [Colon]
1             [Colon]
2             [Colon]
3             [Colon]
4             [Colon]
                ...
```

API column conversion

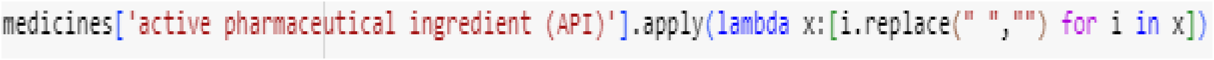


```
medicines['active pharmaceutical ingredient (API)'].apply(lambda x:x.split())
```

2- Get rid of [white spaces]

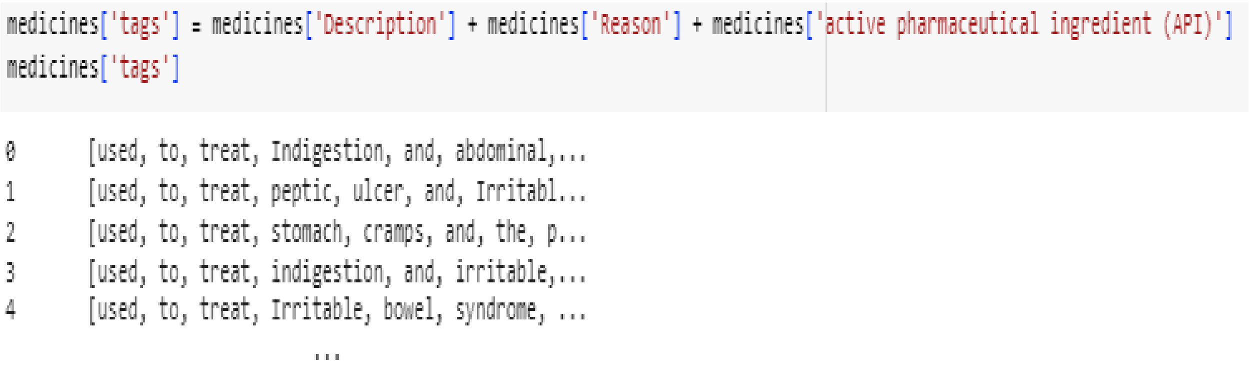


```
medicines['Description'] = medicines['Description'].apply(lambda x:[i.replace(" ","") for i in x])
```



```
medicines['active pharmaceutical ingredient (API)'].apply(lambda x:[i.replace(" ","") for i in x])
```

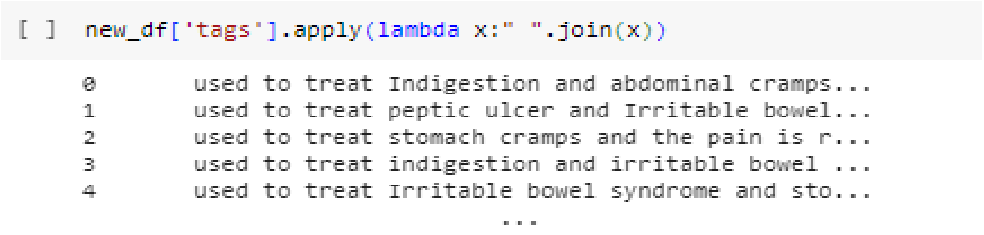
3- Creating the Tag of comparison: -



```
medicines['tags'] = medicines['Description'] + medicines['Reason'] + medicines['active pharmaceutical ingredient (API)']
medicines['tags']
```

```
0       [used, to, treat, Indigestion, and, abdominal,...
1       [used, to, treat, peptic, ulcer, and, Irritabl...
2       [used, to, treat, stomach, cramps, and, the, p...
3       [used, to, treat, indigestion, and, irritable,...
4       [used, to, treat, Irritable, bowel, syndrome, ...
                        ...
```

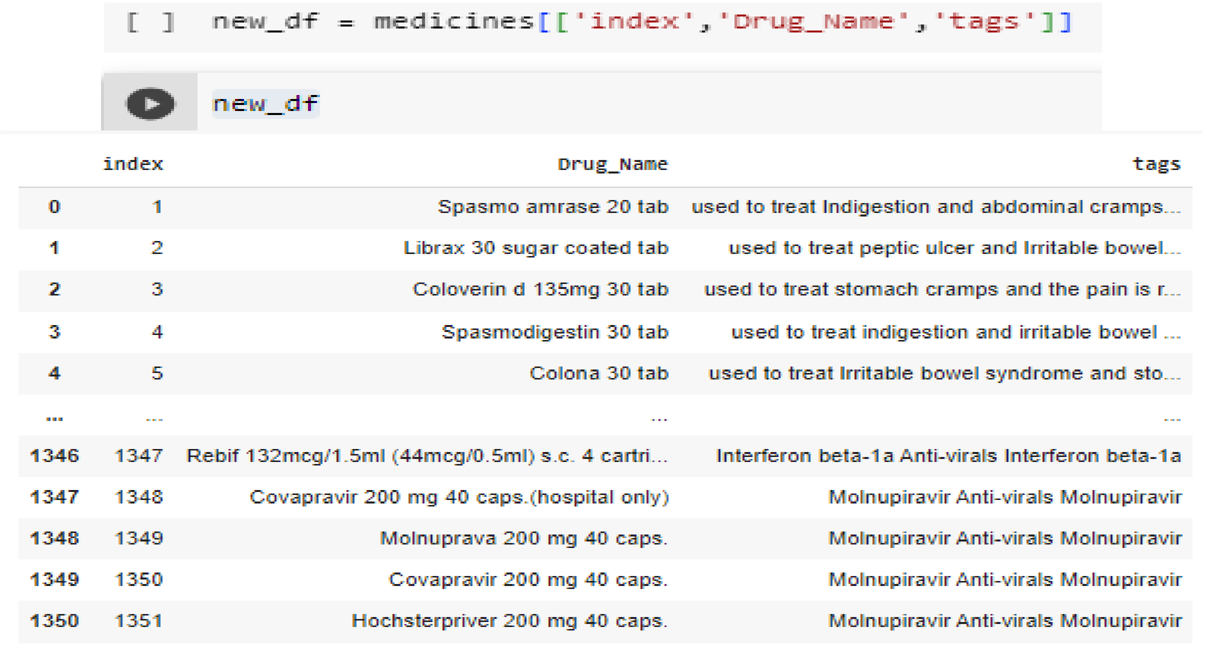
4- Convert Tags into the string format: -



```
[ ]  new_df['tags'].apply(lambda x:" ".join(x))
```

```
0       used to treat Indigestion and abdominal cramps...
1       used to treat peptic ulcer and Irritable bowel...
2       used to treat stomach cramps and the pain is r...
3       used to treat indigestion and irritable bowel ...
4       used to treat Irritable bowel syndrome and sto...
                        ...
```

5- Create new data Frame with [index, drug name, Tag]: -



```
[ ]  new_df = medicines[['index','Drug_Name','tags']]
```

```
new_df
```

	index	Drug_Name	tags
0	1	Spasmo amrase 20 tab	used to treat Indigestion and abdominal cramps...
1	2	Librax 30 sugar coated tab	used to treat peptic ulcer and Irritable bowel...
2	3	Coloverin d 135mg 30 tab	used to treat stomach cramps and the pain is r...
3	4	Spasmodigestin 30 tab	used to treat indigestion and irritable bowel ...
4	5	Colona 30 tab	used to treat Irritable bowel syndrome and sto...
...	...	...	...
1346	1347	Rebif 132mcg/1.5ml (44mcg/0.5ml) s.c. 4 cartri...	Interferon beta-1a Anti-virals Interferon beta-1a
1347	1348	Covapravir 200 mg 40 caps.(hospital only)	Molnupiravir Anti-virals Molnupiravir
1348	1349	Molnuprava 200 mg 40 caps.	Molnupiravir Anti-virals Molnupiravir
1349	1350	Covapravir 200 mg 40 caps.	Molnupiravir Anti-virals Molnupiravir
1350	1351	Hochsterpriver 200 mg 40 caps.	Molnupiravir Anti-virals Molnupiravir

2.1.4- Data Conversion: -

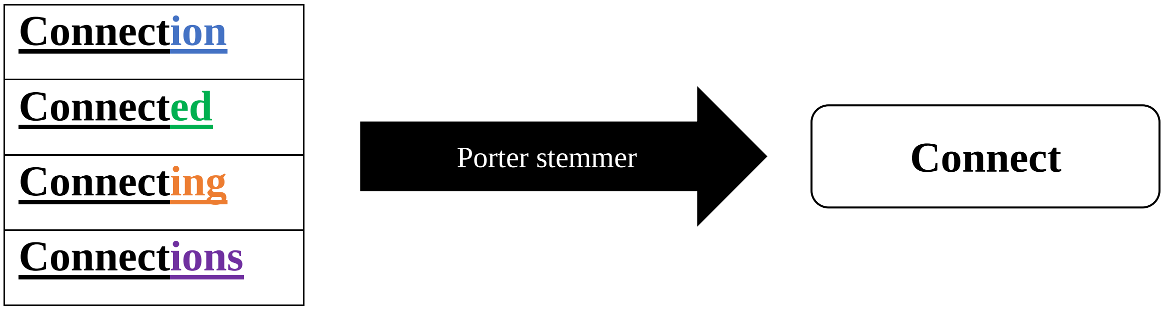
The suitable format to deal with cosine similarity algorithm which (represents the model that classifies recommended medicines) is converting the nominal data into numerical data by passing through two Operations porter stammer and count vectorization.

2.1.4.1-Porter Stemmer: -

1- Definition: -

It’s a process of removing **suffix** from the end of words to get its root.

2- Examples: -



Connection
Connected
Connecting
Connections

3- Benefits: -

- Make the words in the system less complex.
- Reduce the size and capacity of the systems.

4- Operation: -

The operation of the porter stemmer is based on the general formula of word composition and the conditions of removing suffix.

1-The general formula of a word

$$[C]\ ((VC)^{m}\ [V]$$

Where: - C: is consonant string

V: is vowel string

M: measure of full or partial word in form of VC

Example: -

TREE	TROUBLE	TROUBLES
ccvv	ccvvccv	ccvvccvc
M=0	M=1	M=2

2-The Condition of removing suffix

- The general format: -

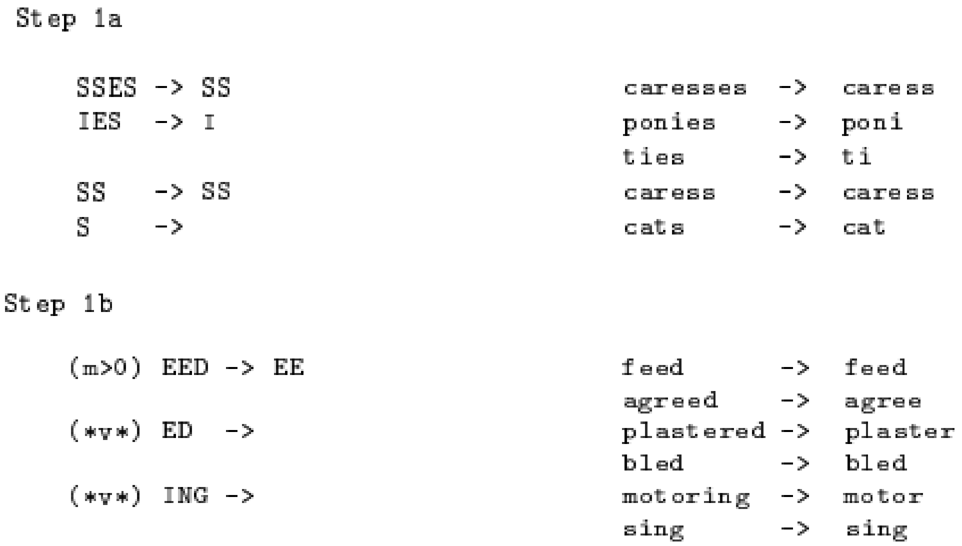
(Condition) Suffix1→replaced to→Sufiix2

Example: -

Condition	word	Root word
(M>1) (S1=ement) (S2=null)	replacement	replac
(M>1) (S1=ement) (S2=null)	cement	cement

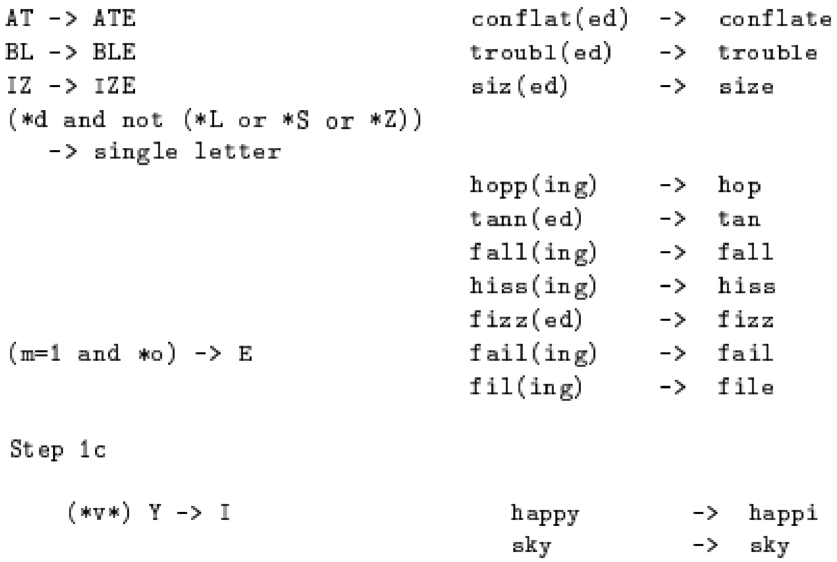
- The steps of algorithms based on suffix removing format: -

Step 1: deals with plurals and past participles ends of words.



```
Step 1a

    SSES -> SS                    caresses  ->  caress
    IES  -> I                     ponies    ->  poni
                                  ties      ->  ti
    SS   -> SS                    caress    ->  caress
    S    ->                       cats      ->  cat

Step 1b

    (m>0) EED -> EE               feed      ->  feed
                                  agreed    ->  agree
    (*v*) ED  ->                  plastered ->  plaster
                                  bled      ->  bled
    (*v*) ING ->                  motoring  ->  motor
                                  sing      ->  sing
```

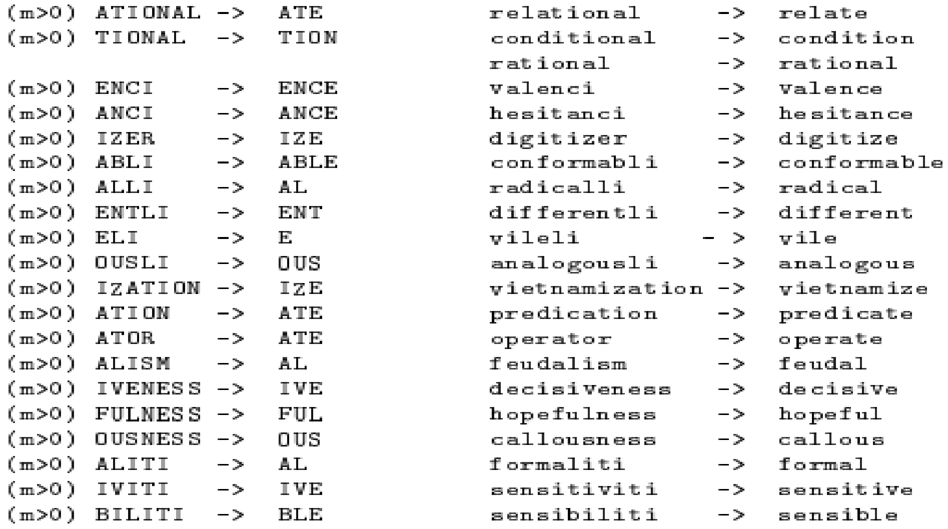
Note: if the 2nd or 3rd rules in step 1b successful the following is done



```
AT -> ATE                   conflat(ed)  ->  conflate
BL -> BLE                   troubl(ed)   ->  trouble
IZ -> IZE                   siz(ed)      ->  size
(*d and not (*L or *S or *Z))
   -> single letter
                            hopp(ing)    ->  hop
                            tann(ed)     ->  tan
                            fall(ing)    ->  fall
                            hiss(ing)    ->  hiss
                            fizz(ed)     ->  fizz
(m=1 and *o) -> E           fail(ing)    ->  fail
                            fil(ing)     ->  file

Step 1c

    (*v*) Y -> I              happy        ->  happi
                              sky          ->  sky
```

Fig23: Step 1 of porter Stemmer algorithm.

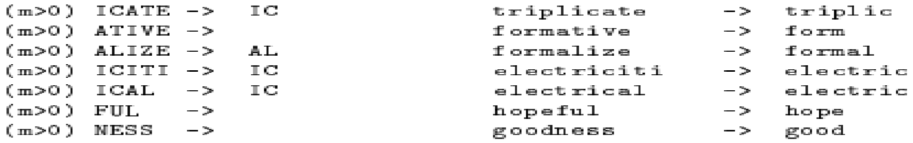
Step 2:



(m>0)	ATIONAL	->	ATE	relational	->	relate
(m>0)	TIONAL	->	TION	conditional	->	condition
				rational	->	rational
(m>0)	ENCI	->	ENCE	valenci	->	valence
(m>0)	ANCI	->	ANCE	hesitanci	->	hesitance
(m>0)	IZER	->	IZE	digitizer	->	digitize
(m>0)	ABLI	->	ABLE	conformabli	->	conformable
(m>0)	ALLI	->	AL	radicalli	->	radical
(m>0)	ENTLI	->	ENT	differentli	->	different
(m>0)	ELI	->	E	vileli	- >	vile
(m>0)	OUSLI	->	OUS	analogousli	->	analogous
(m>0)	IZATION	->	IZE	vietnamization	->	vietnamize
(m>0)	ATION	->	ATE	predication	->	predicate
(m>0)	ATOR	->	ATE	operator	->	operate
(m>0)	ALISM	->	AL	feudalism	->	feudal
(m>0)	IVENESS	->	IVE	decisiveness	->	decisive
(m>0)	FULNESS	->	FUL	hopefulness	->	hopeful
(m>0)	OUSNESS	->	OUS	callousness	->	callous
(m>0)	ALITI	->	AL	formaliti	->	formal
(m>0)	IVITI	->	IVE	sensitiviti	->	sensitive
(m>0)	BILITI	->	BLE	sensibiliti	->	sensible

Fig24: Step 2 of porter Stemmer algorithm.

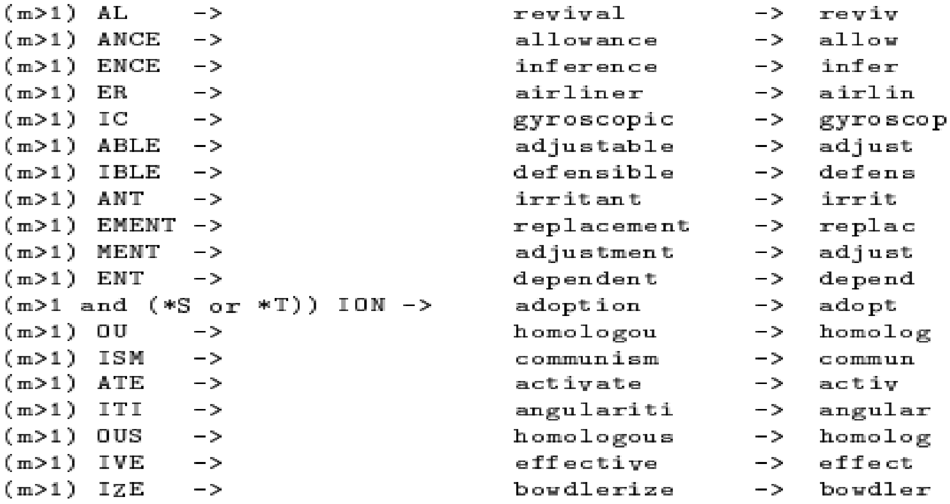
Step 3:



(m>0)	ICATE	->	IC	triplicate	->	triplic
(m>0)	ATIVE	->		formative	->	form
(m>0)	ALIZE	->	AL	formalize	->	formal
(m>0)	ICITI	->	IC	electriciti	->	electric
(m>0)	ICAL	->	IC	electrical	->	electric
(m>0)	FUL	->		hopeful	->	hope
(m>0)	NESS	->		goodness	->	good

Fig25: Step 3 of porter Stemmer algorithm.

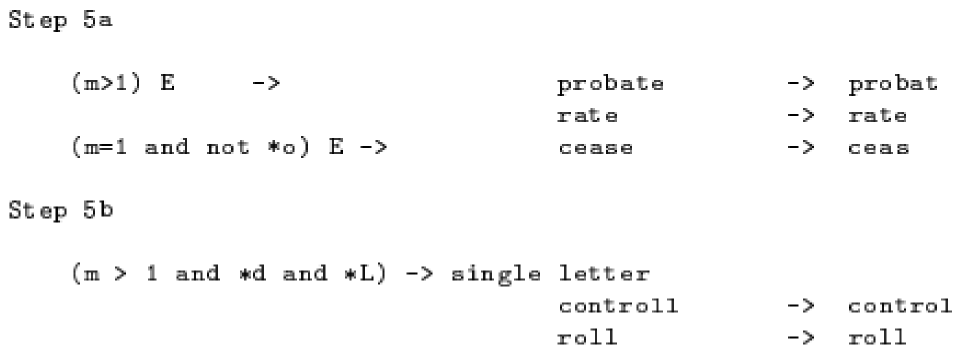
Step 4:



(m>1)	AL	->	revival	->	reviv
(m>1)	ANCE	->	allowance	->	allow
(m>1)	ENCE	->	inference	->	infer
(m>1)	ER	->	airliner	->	airlin
(m>1)	IC	->	gyroscopic	->	gyroscop
(m>1)	ABLE	->	adjustable	->	adjust
(m>1)	IBLE	->	defensible	->	defens
(m>1)	ANT	->	irritant	->	irrit
(m>1)	EMENT	->	replacement	->	replac
(m>1)	MENT	->	adjustment	->	adjust
(m>1)	ENT	->	dependent	->	depend
(m>1 and (*S or *T))	ION	->	adoption	->	adopt
(m>1)	OU	->	homologou	->	homolog
(m>1)	ISM	->	communism	->	commun
(m>1)	ATE	->	activate	->	activ
(m>1)	ITI	->	angulariti	->	angular
(m>1)	OUS	->	homologous	->	homolog
(m>1)	IVE	->	effective	->	effect
(m>1)	IZE	->	bowdlerize	->	bowdler

Fig26: Step 4 of porter Stemmer algorithm.

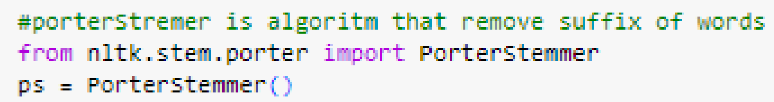
Step 5:



```
Step 5a

    (m>1) E     ->                    probate        ->  probat
                                      rate           ->  rate
    (m=1 and not *o) E ->             cease          ->  ceas

Step 5b

    (m > 1 and *d and *L) -> single letter
                                      controll       ->  control
                                      roll           ->  roll
```

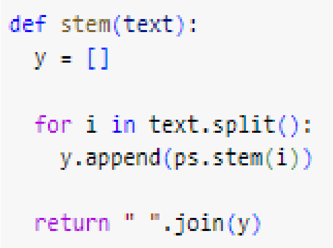
Fig26: Step 5 of porter Stemmer algorithm.

5- Porter stemmer in our model: -



```
#porterStremer is algoritm that remove suffix of words
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

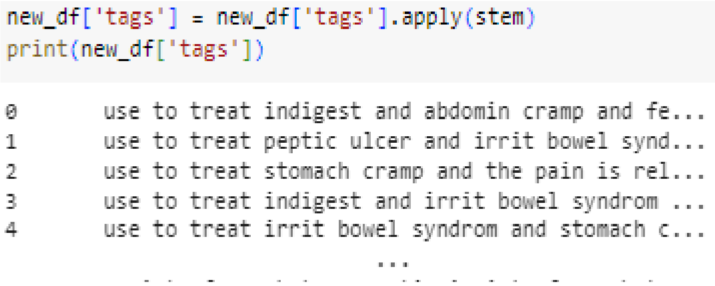
Load porter stemmer



```
def stem(text):
  y = []

  for i in text.split():
    y.append(ps.stem(i))

  return " ".join(y)
```

Function to split the text into wrords.

apply porter stemmer on them



```
new_df['tags'] = new_df['tags'].apply(stem)
print(new_df['tags'])
```

```
0       use to treat indigest and abdomin cramp and fe...
1       use to treat peptic ulcer and irrit bowel synd...
2       use to treat stomach cramp and the pain is rel...
3       use to treat indigest and irrit bowel syndrom ...
4       use to treat irrit bowel syndrom and stomach c...
                              ...
```

The tags of all medicines with roots words

2.1.4.2-Count Vectorization: -

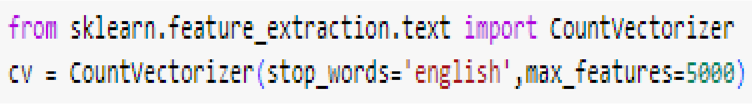
1- Definition: -

It is a process of converting text to a vector of numerical data based on the number of repetitions of each word.

2- Example: -

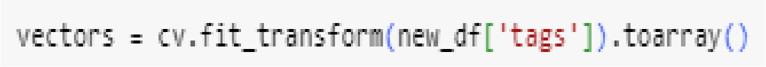
Phrases (medicine info)	treatment	of	colon	ulcer	indigestion
Treatment of colon, ulcer	1	1	1	1	0
Treatment of ulcer, indigestion	1	1	0	1	1

3- Count vectorizer into model: -



```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(stop_words='english',max_features=5000)
```

Import the count vectorizer.



```
vectors = cv.fit_transform(new_df['tags']).toarray()
```

Apply the count vectorizer on the Vectors.

2.1.5- Machine Learning Model: -

2.1.5.1- Introduction: -

The model we will use for medicines recommendations is the cosine similarity algorithm.

Cosine similarity is a measure of similarity between two data points in a plane, so it is used for determining the distance between the neighbors.

2.1.5.2- Operation: -

Cosine similarity is the cosine of the angle between two vectors, it is used as a distance evaluation metric between two points in the plane. The increase in distance the similarity of data points reduces.

There are two methods of cosine similarity process the first one is graphically and dealing with words representation on axis and not practically but it's easy to understand and the other is mathematically, it's more complex but more practical.

1- Graphically: -

1- create the table of vectorized phrases

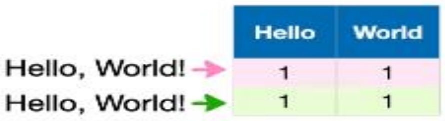
2- plot the points of words on the graph

3- draw the lines from origin to points

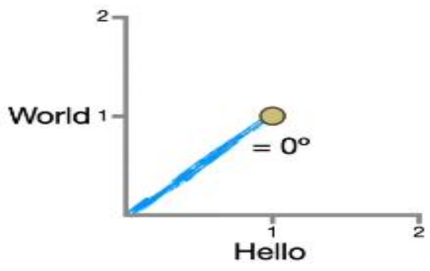
4- calculate the angle between them

5-the Similarity of the phrases equals Cosine (angle)

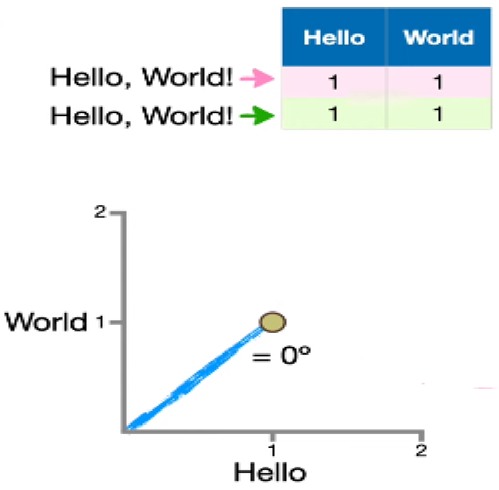
Examples



	Hello	World
Hello, World! →	1	1
Hello, World! →	1	1

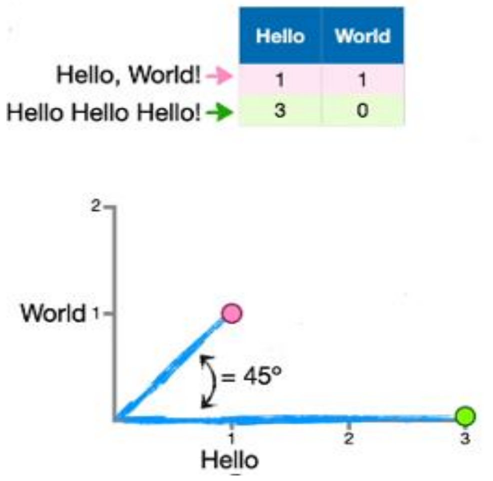


in case of identical phrases
1- angle =0
2- similarity = cos(0) = 1



	Hello	World
Hello, World!	1	1
Hello, World!	1	1

in case of different phrases
1- angle =90
2- similarity = cos(90) = zero



	Hello	World
Hello, World!	1	1
Hello Hello Hello!	3	0

in case of similar phrases
1- angle = [0,90]
2- similarity = cos(angle) = [0,1]
3- does not effected by repetition of words

cos(45°) = 0.71 = Cosine Similarity

The number of words ∝ the number of dimensions of Graph

2- Mathematical: -

It is used in case of comparison between phrases of more than 3 words.

1- create the table of vectorized phrases

2- use the cosine similarity formula

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Example (1)

	hello	word
X=Hello world	1	1
Y=Hello Hello Hello	1	0

$$\text{Cosine similarity} = = \frac{(1*1)+(1*0)}{\sqrt{1^2 +1^2} +\sqrt{1^2 +0^2}} = 0.71$$

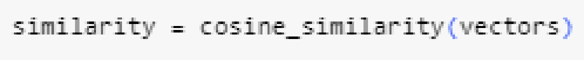
Example (2)

Phrases (medicine info)	treatment	of	colon	ulcer	indigestion
Treatment of colon, ulcer	1	1	1	1	0
Treatment of ulcer, indigestion	1	1	0	1	1

$$\text{Cosine similarity} = = \frac{(1*1)+(1*1)+(1*0)+\left(1*1+(0*1)\right)}{\sqrt{1^2 +1^2+1^2 +1^2 +0^2} +\sqrt{1^2 +1^2 +0^2+1^2 +1^2}} = .75$$

2.1.5.3- Algorithm of cosine similarity: -

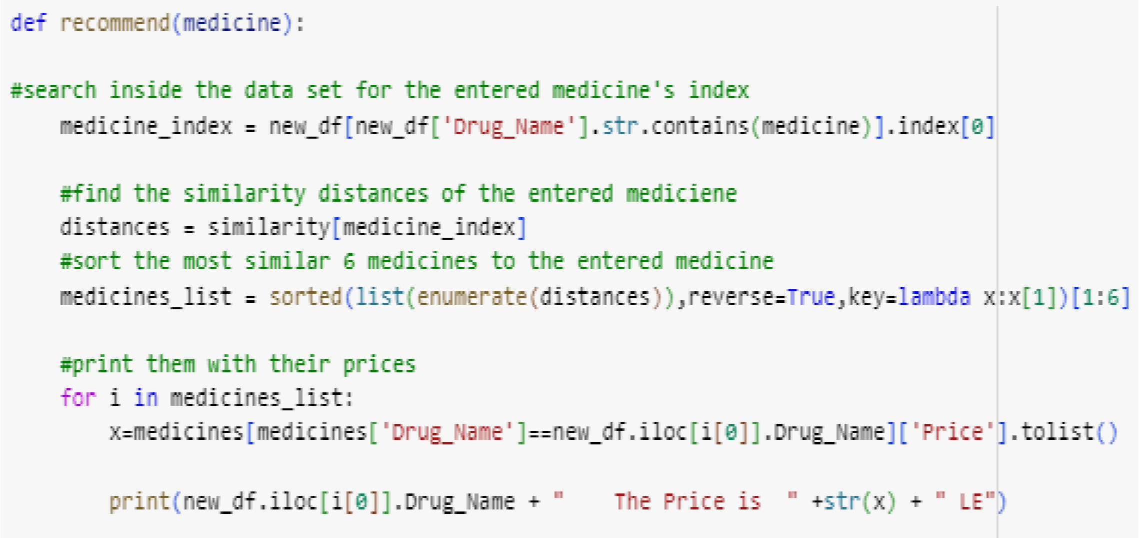
1- calculate the similarity among all medicines: -



```
similarity = cosine_similarity(vectors)
```

Note: vectors are the numerical format of medicines TAG dataset

2- Recommendation function: -



```
def recommend(medicine):

#search inside the data set for the entered medicine's index
    medicine_index = new_df[new_df['Drug_Name'].str.contains(medicine)].index[0]

    #find the similarity distances of the entered mediciene
    distances = similarity[medicine_index]
    #sort the most similar 6 medicines to the entered medicine
    medicines_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

    #print them with their prices
    for i in medicines_list:
        x=medicines[medicines['Drug_Name']==new_df.iloc[i[0]].Drug_Name]['Price'].tolist()

        print(new_df.iloc[i[0]].Drug_Name + "    The Price is  " +str(x) + " LE")
```

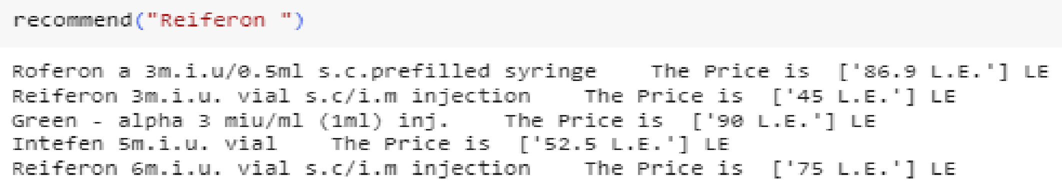
Medicine Recommendation function

1- search about the entered medicine’s and get its index of it in dataset

2- get the similarity of that medicine’s index

3- sort the highest five similar medicines to the entered medicine

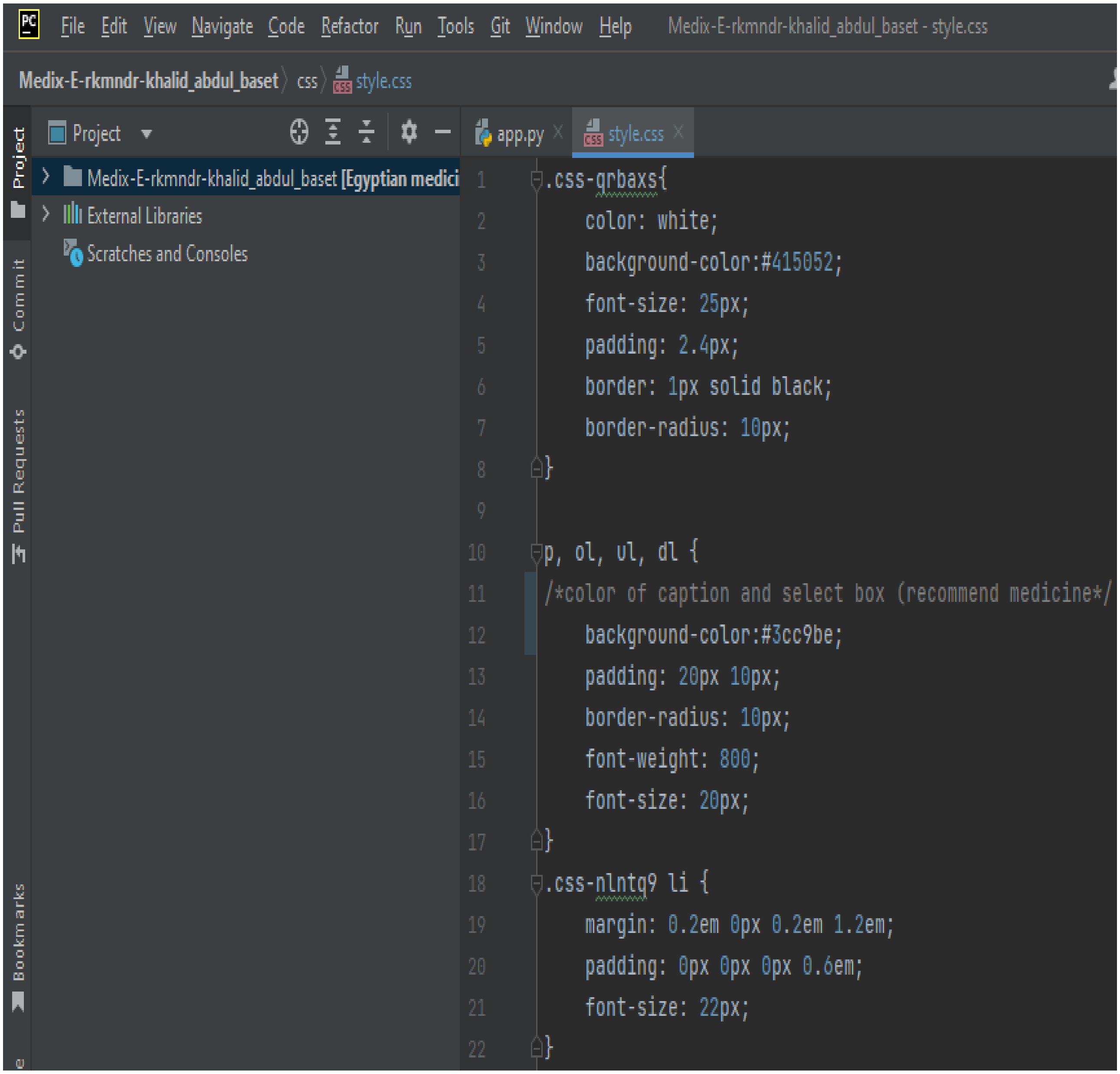
4- print them with their prices



```
recommend("Reiferon ")

Roferon a 3m.i.u/0.5ml s.c.prefilled syringe    The Price is  ['86.9 L.E.'] LE
Reiferon 3m.i.u. vial s.c/i.m injection    The Price is  ['45 L.E.'] LE
Green - alpha 3 miu/ml (1ml) inj.    The Price is  ['90 L.E.'] LE
Intefen 5m.i.u. vial    The Price is  ['52.5 L.E.'] LE
Reiferon 6m.i.u. vial s.c/i.m injection    The Price is  ['75 L.E.'] LE
```

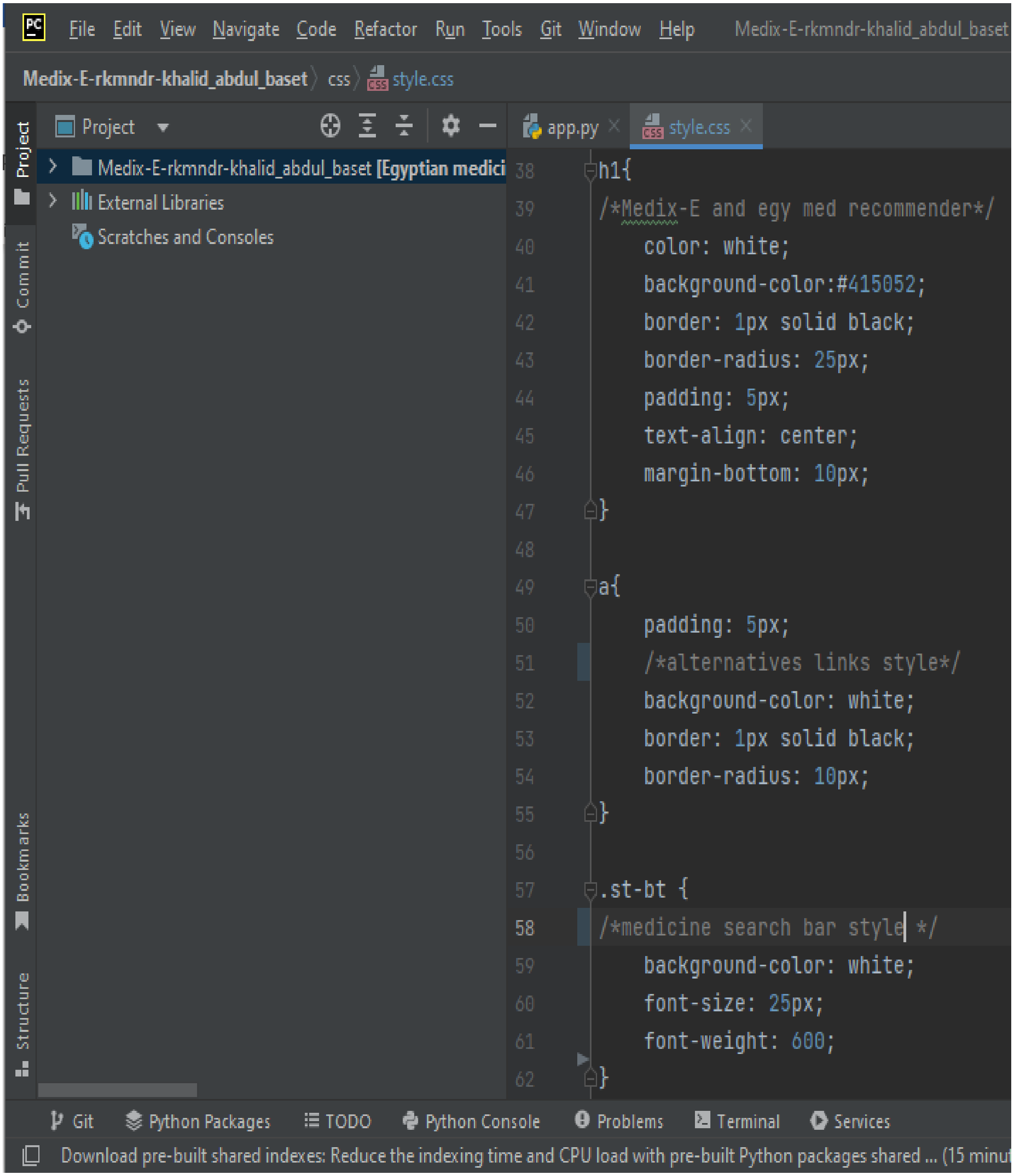
2.2- Front-End Design of Web Application

2.2.1- Styles (CSS file): -


PC
File Edit View Navigate Code Refactor Run Tools Git Window Help
Medix-E-rkmndr-khalid_abdul_baset - style.css
Medix-E-rkmndr-khalid_abdul_baset 〉 css 〉 style.css
Project
Medix-E-rkmndr-khalid_abdul_baset [Egyptian medici
External Libraries
Scratches and Consoles
app.py
style.css
Project
Commit
Pull Requests
Bookmarks

```css
.css-grbaxs{
    color: white;
    background-color:#415052;
    font-size: 25px;
    padding: 2.4px;
    border: 1px solid black;
    border-radius: 10px;
}

p, ol, ul, dl {
/*color of caption and select box (recommend medicine*/
    background-color:#3cc9be;
    padding: 20px 10px;
    border-radius: 10px;
    font-weight: 800;
    font-size: 20px;
}
.css-nlntq9 li {
    margin: 0.2em 0px 0.2em 1.2em;
    padding: 0px 0px 0px 0.6em;
    font-size: 22px;
}
```

Fig26: Styles of select box and caption box.

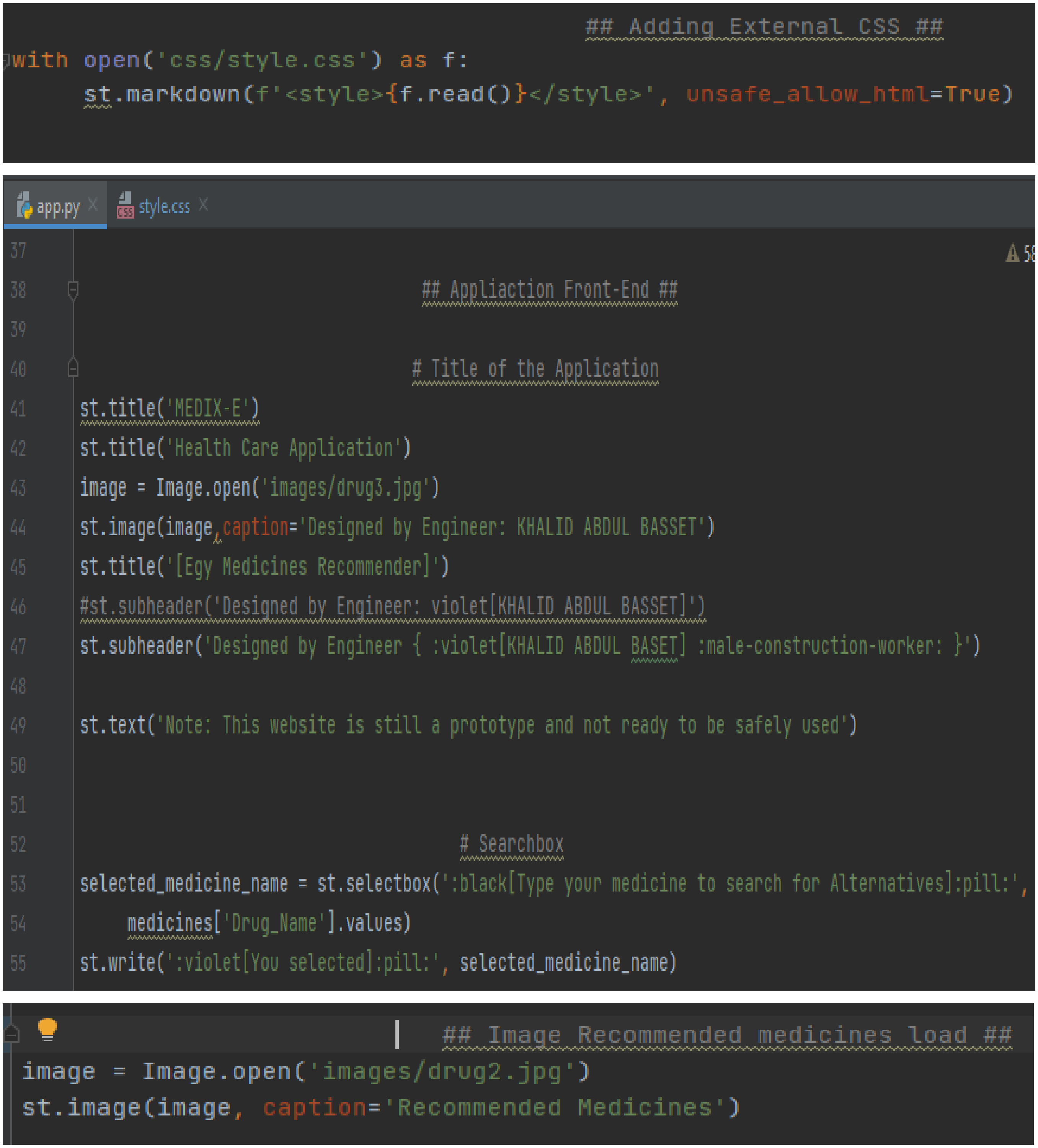


```
h1{
/*Medix-E and egy med recommender*/
    color: white;
    background-color:#415052;
    border: 1px solid black;
    border-radius: 25px;
    padding: 5px;
    text-align: center;
    margin-bottom: 10px;
}

a{

    padding: 5px;
    /*alternatives links style*/
    background-color: white;
    border: 1px solid black;
    border-radius: 10px;
}

.st-bt {
/*medicine search bar style */
    background-color: white;
    font-size: 25px;
    font-weight: 600;
}
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared ... (15 minut

Fig27: Styles of header, search bar box and medicines' links box

2.2.2- python file: -



```
                                        ## Adding External CSS ##
with open('css/style.css') as f:
    st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)
```

```

                         ## Appliaction Front-End ##

                         # Title of the Application
st.title('MEDIX-E')
st.title('Health Care Application')
image = Image.open('images/drug3.jpg')
st.image(image,caption='Designed by Engineer: KHALID ABDUL BASSET')
st.title('[Egy Medicines Recommender]')
#st.subheader('Designed by Engineer: violet[KHALID ABDUL BASSET]')
st.subheader('Designed by Engineer { :violet[KHALID ABDUL BASET] :male-construction-worker: }')

st.text('Note: This website is still a prototype and not ready to be safely used')



                              # Searchbox
selected_medicine_name = st.selectbox(':black[Type your medicine to search for Alternatives]:pill:',
    medicines['Drug_Name'].values)
st.write(':violet[You selected]:pill:', selected_medicine_name)
```

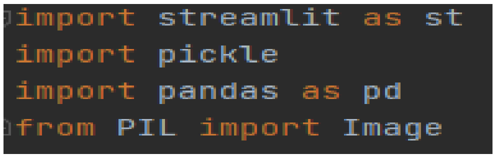
```
                            ## Image Recommended medicines load ##
image = Image.open('images/drug2.jpg')
st.image(image, caption='Recommended Medicines')
```

Fig28: python Front-End design file

2.3- Back-End

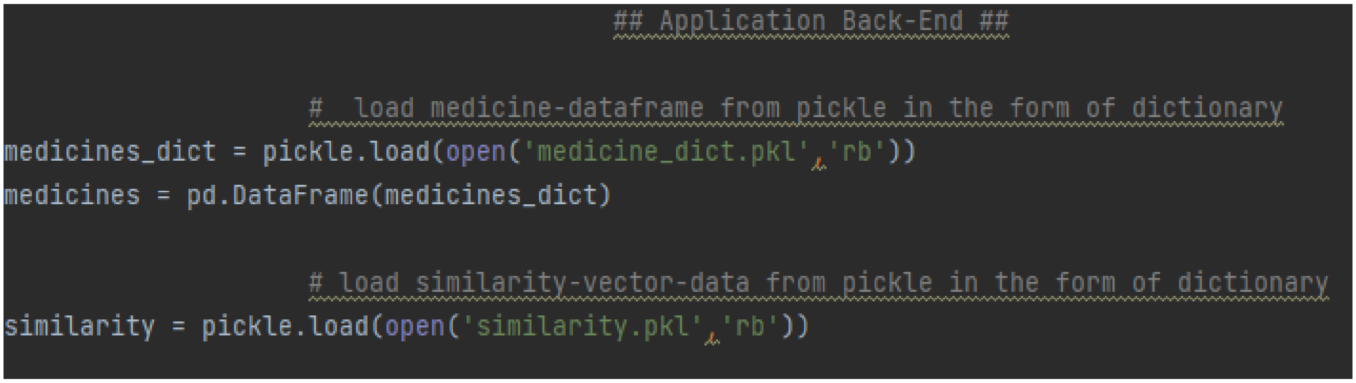
2.3.1- Design the web application: -

2.3.1.1- python file: -



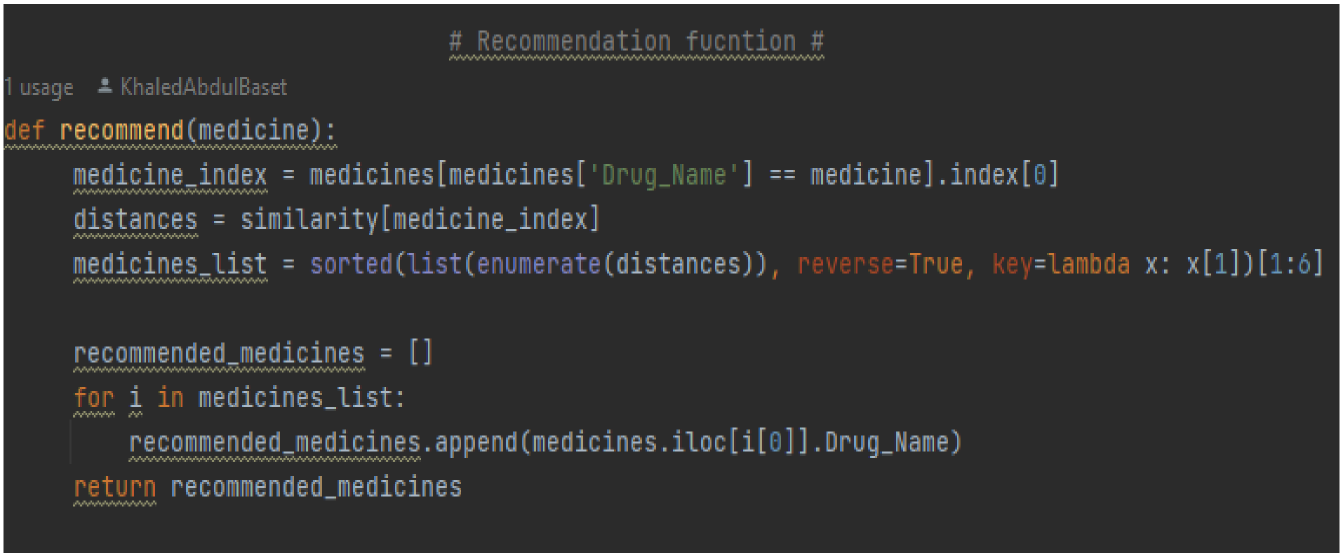
```python
import streamlit as st
import pickle
import pandas as pd
from PIL import Image
```

Import the necessary libraries.



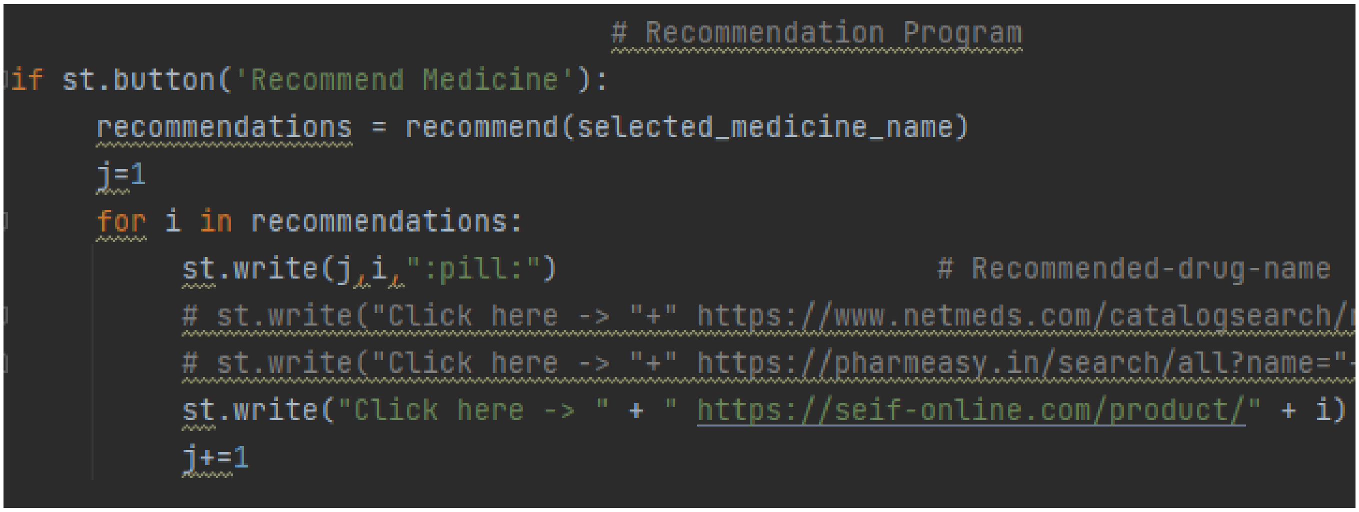
```python
                                    ## Application Back-End ##

                  #  load medicine-dataframe from pickle in the form of dictionary
medicines_dict = pickle.load(open('medicine_dict.pkl','rb'))
medicines = pd.DataFrame(medicines_dict)

                  # load similarity-vector-data from pickle in the form of dictionary
similarity = pickle.load(open('similarity.pkl','rb'))
```

Load the model and the medicines dataset.



```python
                              # Recommendation fucntion #
1 usage  KhaledAbdulBaset
def recommend(medicine):
    medicine_index = medicines[medicines['Drug_Name'] == medicine].index[0]
    distances = similarity[medicine_index]
    medicines_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]

    recommended_medicines = []
    for i in medicines_list:
        recommended_medicines.append(medicines.iloc[i[0]].Drug_Name)
    return recommended_medicines
```

The Recommendation function.



```
                                # Recommendation Program
if st.button('Recommend Medicine'):
    recommendations = recommend(selected_medicine_name)
    j=1
    for i in recommendations:
        st.write(j,i,":pill:")                     # Recommended-drug-name
        # st.write("Click here -> "+" https://www.netmeds.com/catalogsearch/r
        # st.write("Click here -> "+" https://pharmeasy.in/search/all?name="-
        st.write("Click here -> " + " https://seif-online.com/product/" + i)
        j+=1
```

The Recommendation displayer.

2.4-Deployment

2.4.1- Introduction: -

2.4.1.1-What is the deployment process:

Imagine that you spend plenty of time and resources on building machine learning models, Ideally, you would want your model to determine if the performance of the model is effective in real-time so that; this is where model deployment comes in.

Deploying a machine learning model means to integrate a machine learning model into an existing production environment where it can take in an input and return an output.

2.4.1.2- The benefit of deployment process:

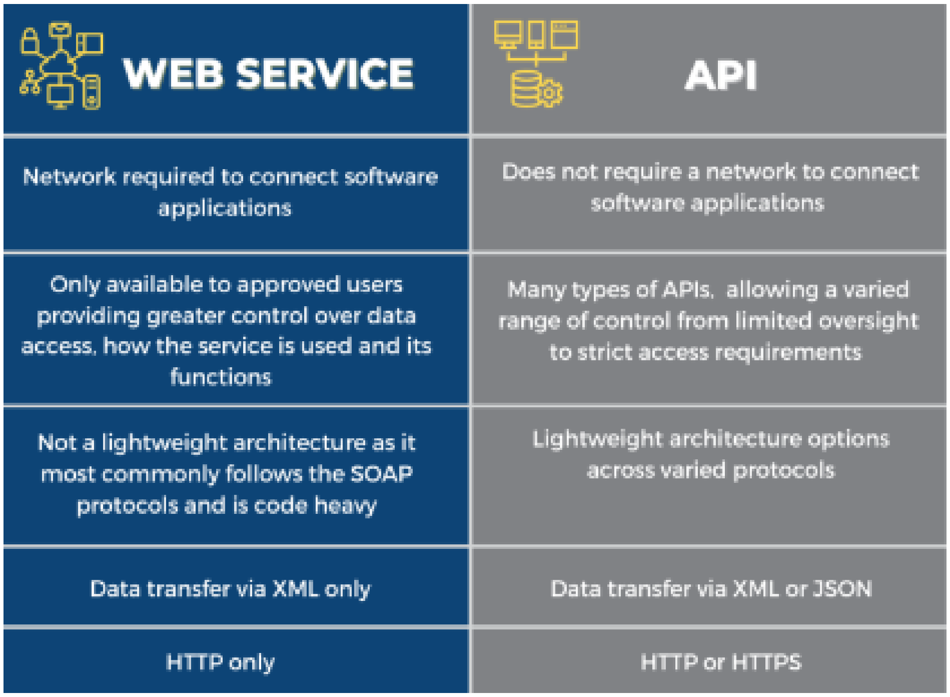
The purpose of deploying our model (medicine recommender) is to make the recommendations from our trained ML model available to others, whether that be users, management, or other systems.

2.3.1.3- Definition of APIs:

APIs (Application Programming Interface) are mechanisms that enable two software components to communicate with each other.

User interface	Api
Connect computer to person	Connect computer to others

2.2.3.4- Difference between web services and APIs:



WEB SERVICE	API
Network required to connect software applications	Does not require a network to connect software applications
Only available to approved users providing greater control over data access, how the service is used and its functions	Many types of APIs, allowing a varied range of control from limited oversight to strict access requirements
Not a lightweight architecture as it most commonly follows the SOAP protocols and is code heavy	Lightweight architecture options across varied protocols
Data transfer via XML only	Data transfer via XML or JSON
HTTP only	HTTP or HTTPS

In our medicine recommendation system, we will deploy our model as a web service that has its own back end design and front end designs.

2.2.3.5- Streamlit:

Streamlit is an open-source app framework in python language. It helps us create beautiful web apps for data science and machine learning in a little time. It is compatible with major python libraries such as scikit-learn, keras, PyTorch, latex, numpy, pandas, matplotlib, etc.

It also allows data scientists to easily create free machine learning applications.

2.4.2 - Operation: -

We aim to make our machine learning web application available for any user, so we need it to be global accessed through a certain link, that is way we will deploy our web applications on Streamlit open-source app framework as it supports the python application.

After deploying the ML web application:

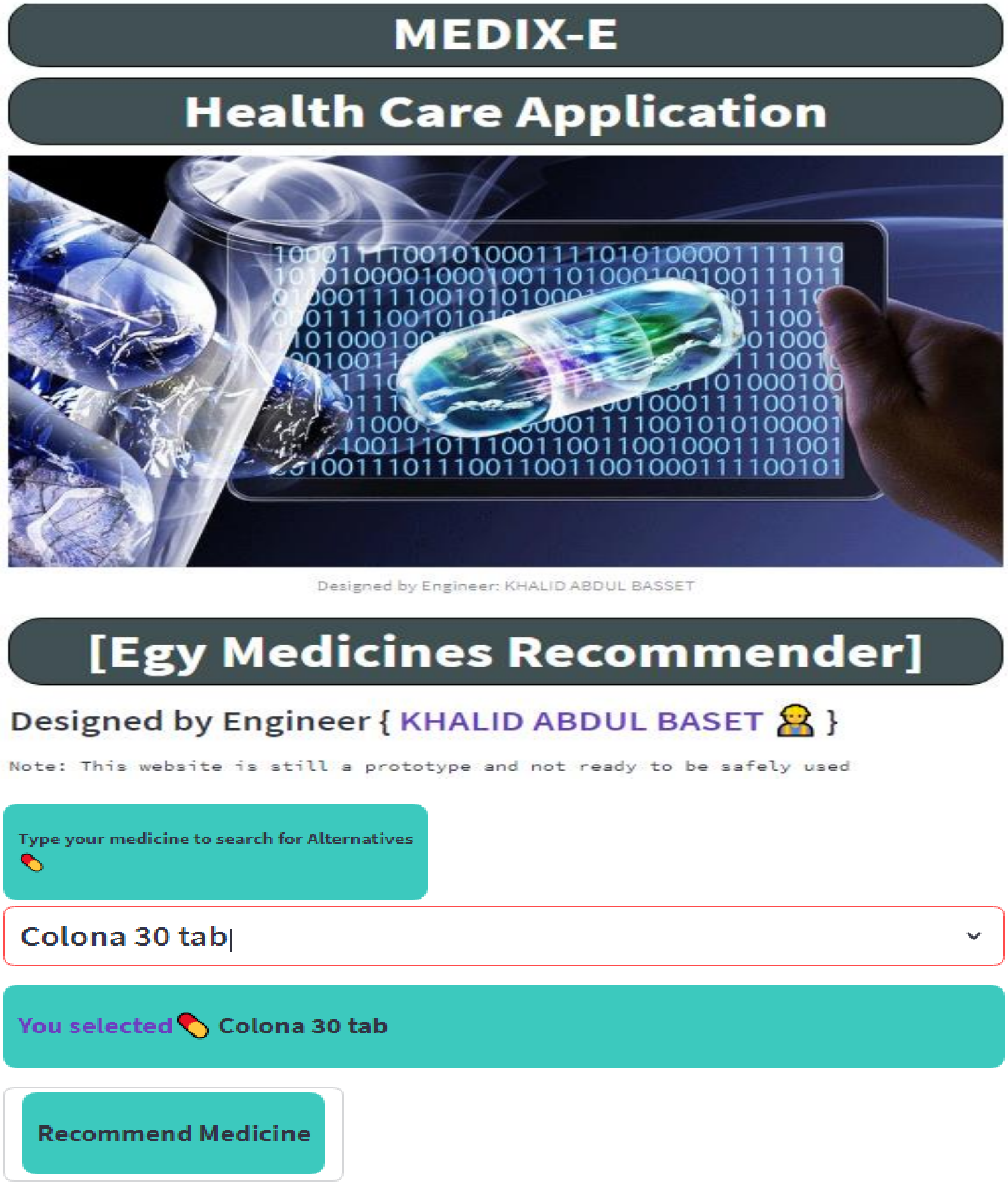
1- we can access it anytime anywhere using a global link.

https://medix-e-rkmndr-kh-basetgi-e4zrdqcfhu.streamlit.app/

2- user can type his described medicine and press recommend to have alternatives of it.

3- the user able to order the alternative medicine from the seif pharmacy via pressing the link.

2.4.3 – Application’s final Form: -



MEDIX-E

Health Care Application

Designed by Engineer: KHALID ABDUL BASSET

[Egy Medicines Recommender]

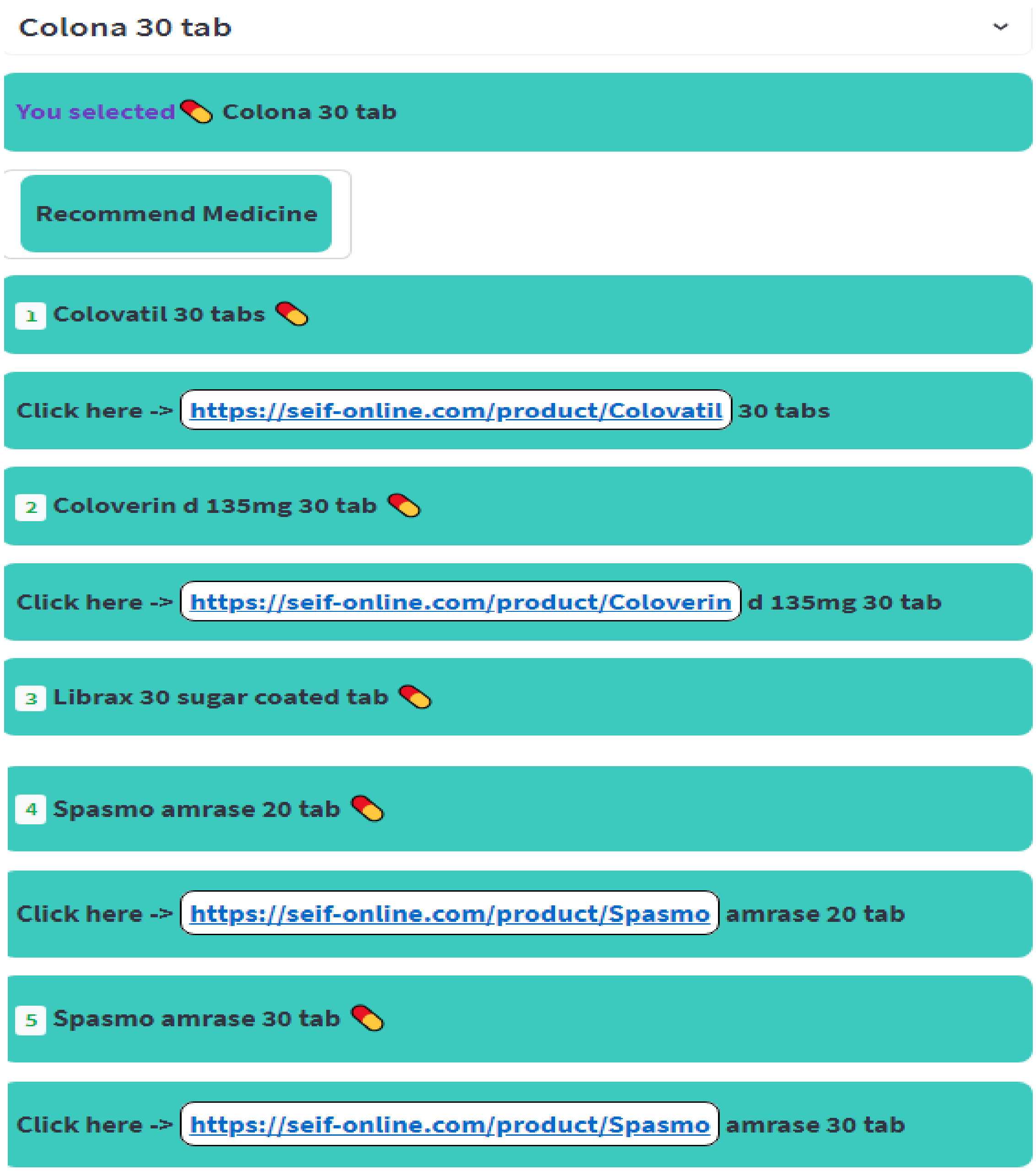
Designed by Engineer { KHALID ABDUL BASET 👷 }

Note: This website is still a prototype and not ready to be safely used

Type your medicine to search for Alternatives 💊

Colona 30 tab

You selected 💊 Colona 30 tab



Colona 30 tab

You selected 💊 Colona 30 tab

Recommend Medicine

1 Colovatil 30 tabs 💊

Click here -> https://seif-online.com/product/Colovatil 30 tabs

2 Coloverin d 135mg 30 tab 💊

Click here -> https://seif-online.com/product/Coloverin d 135mg 30 tab

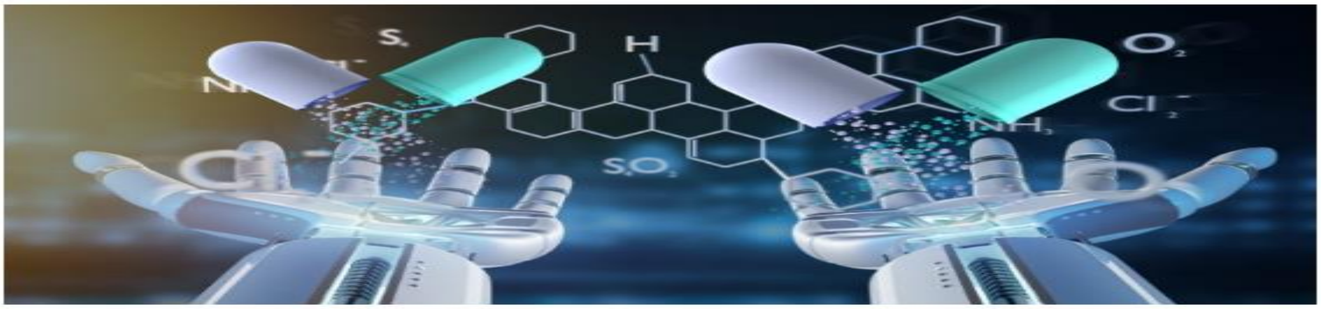
3 Librax 30 sugar coated tab 💊

4 Spasmo amrase 20 tab 💊

Click here -> https://seif-online.com/product/Spasmo amrase 20 tab

5 Spasmo amrase 30 tab 💊

Click here -> https://seif-online.com/product/Spasmo amrase 30 tab



Recommended Medicines

Fig28: Screenshots of Medicine recommender

5- Implementation

Chapter Three: Lab Report detector (covid-19)

1- Introduction: -

In this application, we are going to use logistic regression to predict PCR's (Polymerase chain reaction) result via CBC (complete blood count) values, The application represented as a web page that contains input text, each receives certain value and eventually the application predict the patient's condition (Recovered or Not).

1.1- Problem Statement: -

To check covid 19 accurately you must depend on the PCR which provides greater detail and then find out the condition of the patient. But the main problem is that PCR's Cost very expensive for some segments of populations to test themselves through, the researchers show a relation between PCR and CBC test, knowing that the cost of CBC is very low and suitable for almost all people, from that come out the idea of predicting the result of PCR by using the CBC results.

1.2- Objectives: -

We aim to create an application that spares the patients from the PCR's cost by predicting the results of it using CBC test.

How?

- Create an APP reads the critical information in CBC related to PCR

- Make the Application able to predict the result (Recovered or Not.)

1.3- Motivations: -

This application is dedicated to serve the medical field.

1- Helps patients to diagnosis themselves against covid19.

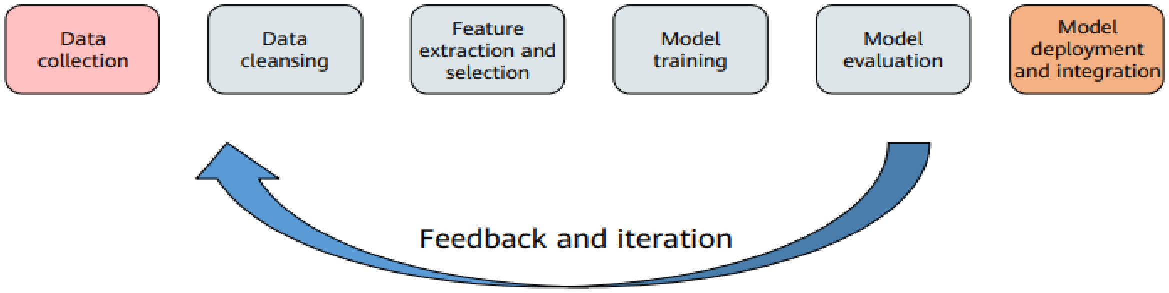
2- Spares the patients from the high cost of PCR.

3- Time and Cost saver

4- More accurate than even the experts

1.4- Development process: -

1- machine leering



The operation of developing any machine learning application must include those processes to make it efficient for use.

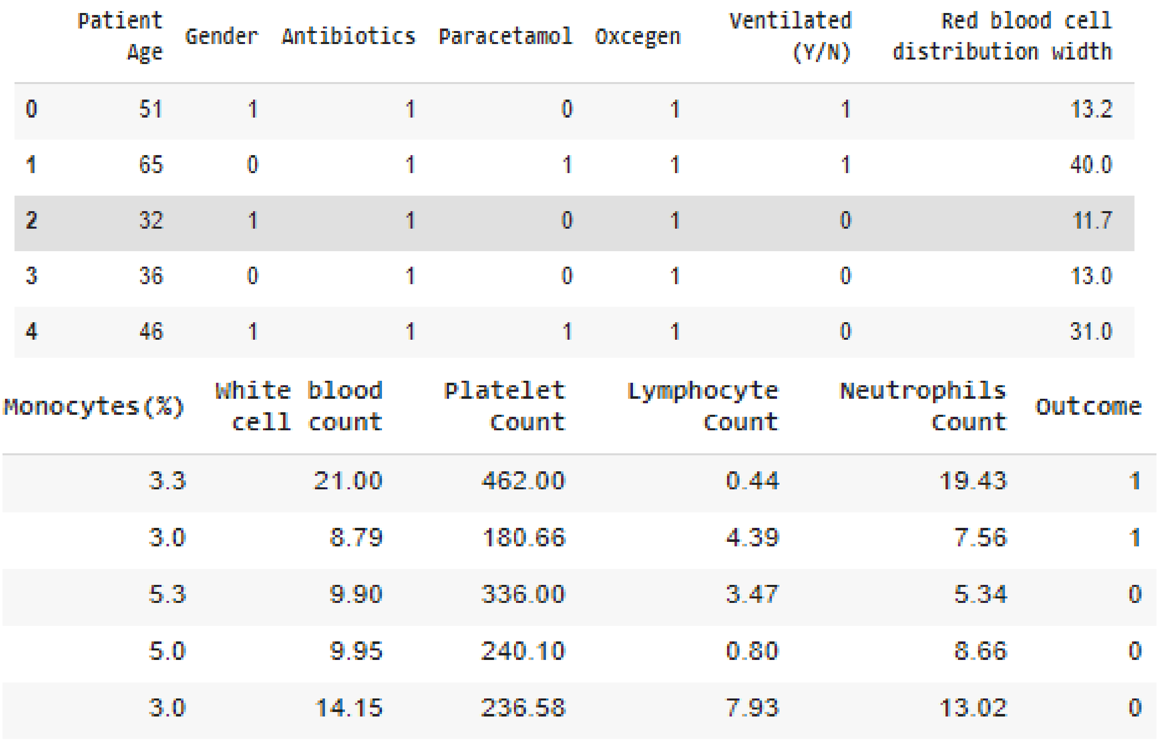
2- Front-End

2- Implementations: -

2.1- Machine Learning

2.1.1- Data Collection: -

A team of researchers and doctors from Qatar University have created a database of complete blood count, demographic data which contains: Age and Gender and the medication course and CBC Information that matters in identifying COVID-19.



	Patient Age	Gender	Antibiotics	Paracetamol	Oxcegen	Ventilated (Y/N)	Red blood cell distribution width	Monocytes(%)	White blood cell count	Platelet Count	Lymphocyte Count	Neutrophils Count	Outcome
0	51	1	1	0	1	1	13.2	3.3	21.00	462.00	0.44	19.43	1
1	65	0	1	1	1	1	40.0	3.0	8.79	180.66	4.39	7.56	1
2	32	1	1	0	1	0	11.7	5.3	9.90	336.00	3.47	5.34	0
3	36	0	1	0	1	0	13.0	5.0	9.95	240.10	0.80	8.66	0
4	46	1	1	1	1	0	31.0	3.0	14.15	236.58	7.93	13.02	0

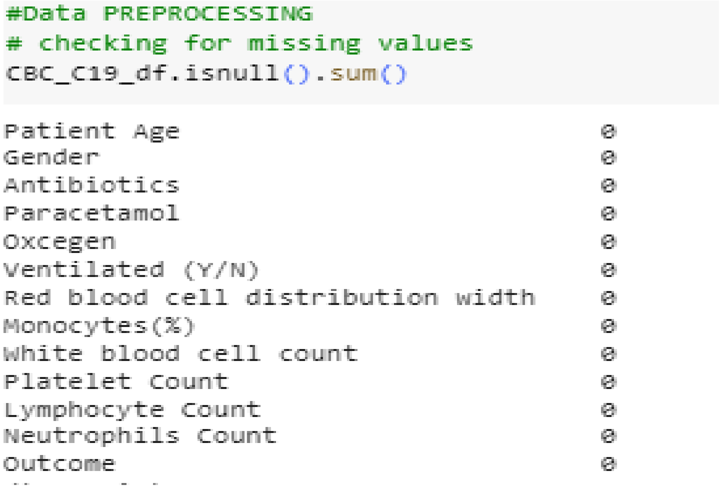
Fig29: Screenshots of Dataset for CBC lab values

The patients' outcome shows whether the patient is recovered or not yet, based on PCR test.

2.1.2- Data Preprocessing (Cleansing): -

2.1.2.1-Processing of lost data: -

Check how many samples contain lost data.

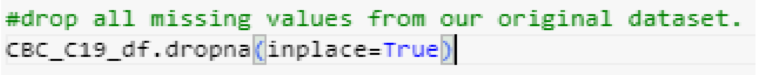


```
#Data PREPROCESSING
# checking for missing values
CBC_C19_df.isnull().sum()
```

```
Patient Age                        0
Gender                             0
Antibiotics                        0
Paracetamol                        0
Oxcegen                            0
Ventilated (Y/N)                   0
Red blood cell distribution width  0
Monocytes(%)                       0
White blood cell count             0
Platelet Count                     0
Lymphocyte Count                   0
Neutrophils Count                  0
Outcome                            0
```

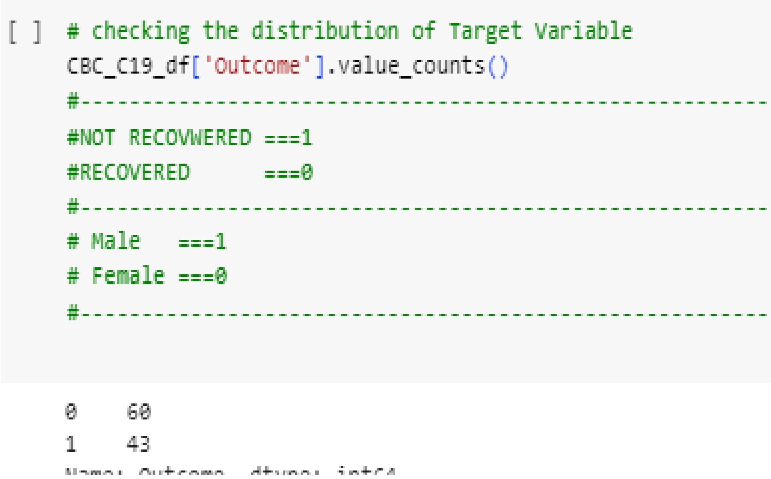
2.1.2.2-Data Filtering: -

Filtering the corrupted data by dropping the samples with missing data



```
#drop all missing values from our original dataset.
CBC_C19_df.dropna(inplace=True)
```

2.1.2.3-Data Notations: -

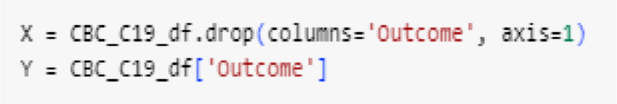


```
[ ]  # checking the distribution of Target Variable
     CBC_C19_df['Outcome'].value_counts()
     #-------------------------------------------------------
     #NOT RECOVWERED ===1
     #RECOVERED      ===0
     #-------------------------------------------------------
     # Male   ===1
     # Female ===0
     #-------------------------------------------------------
```

```
0    60
1    43
Name: Outcome, dtype: int64
```

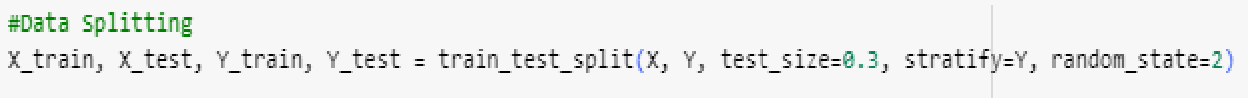
2.1.3- Data Preparing: -

2.1.3.1-Sepration inputs and output: -

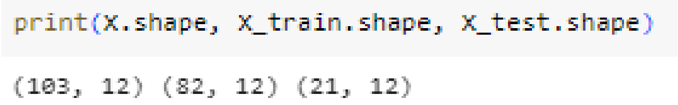


```
X = CBC_C19_df.drop(columns='Outcome', axis=1)
Y = CBC_C19_df['Outcome']
```

2.1.3.2-Data Splitting: -



```
#Data Splitting
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, stratify=Y, random_state=2)
```



```
print(X.shape, X_train.shape, X_test.shape)
```

```
(103, 12) (82, 12) (21, 12)
```

2.1.4- Machine Learning Model: -

2.1.4.1- Import Logistic Regression: -

It is a classification method to predict a binary outcome, such as yes or no, based on prior observations of a data set, used when:

- Data representation doesn't fit linear equation(z=wX+b).

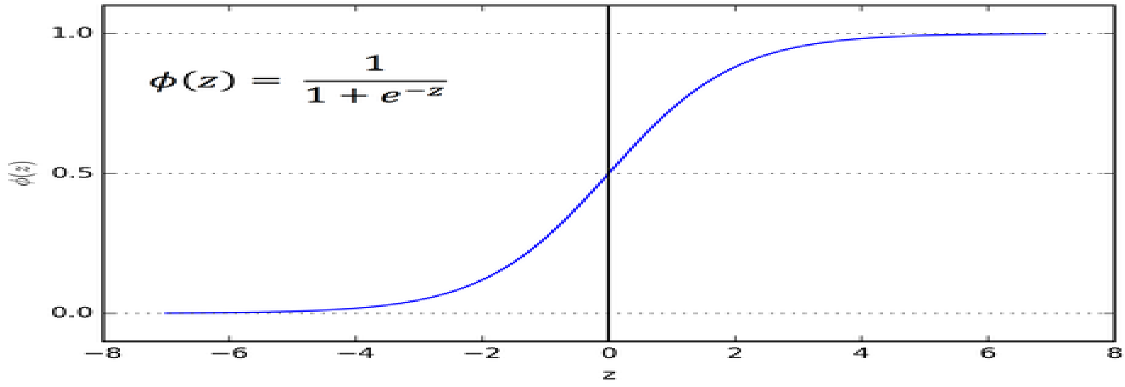


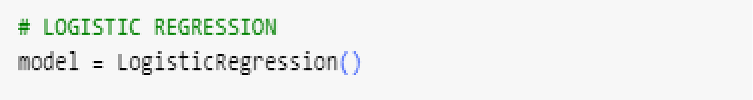
Fig30: Logistic regression curve

The output of logistic regression is between 0 and 1.

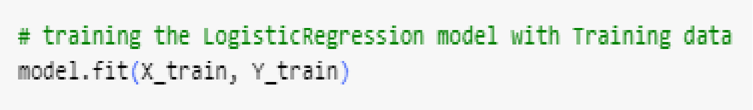
$$\boxed{F(z) = \frac{1}{1+e^{-Z}}}$$

- if z too big(∞), $F(z) = \frac{1}{1+e^{-(-\infty)}} = \frac{1}{1+0} = 1$
- if z too small($-\infty$), $F(z) = \frac{1}{1+e^{-(-\infty)}} = \frac{1}{1+\infty} = 0$

So, the output is in between [0,1]

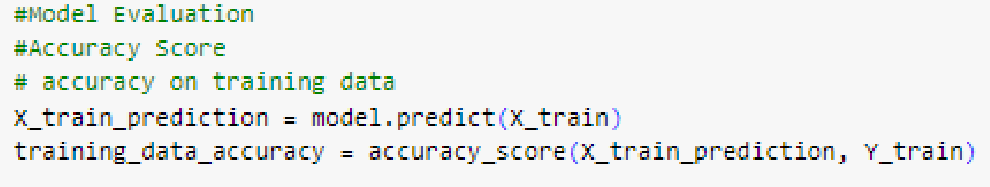


```
# LOGISTIC REGRESSION
model = LogisticRegression()
```

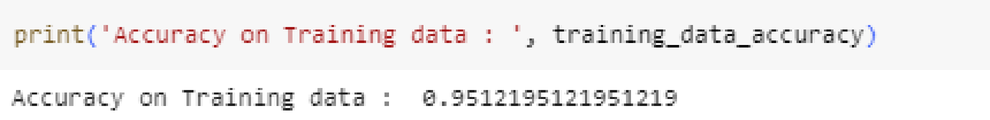


```
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

2.1.4.2- Model Evaluation: -

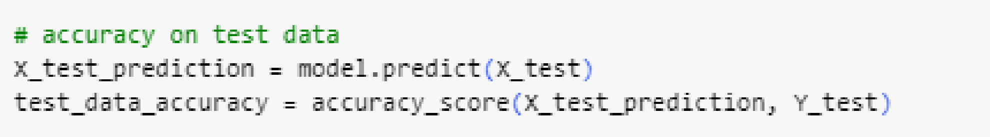


```
#Model Evaluation
#Accuracy Score
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

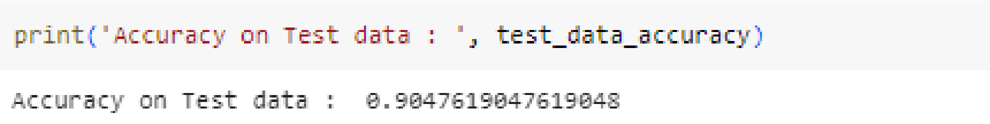


```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.9512195121951219
```



```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```



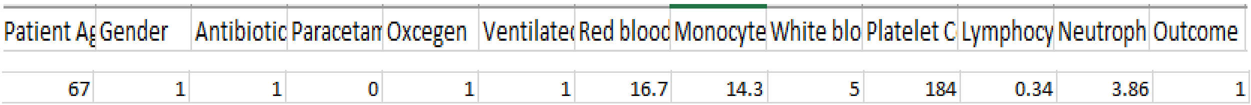
```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.9047619047619048
```

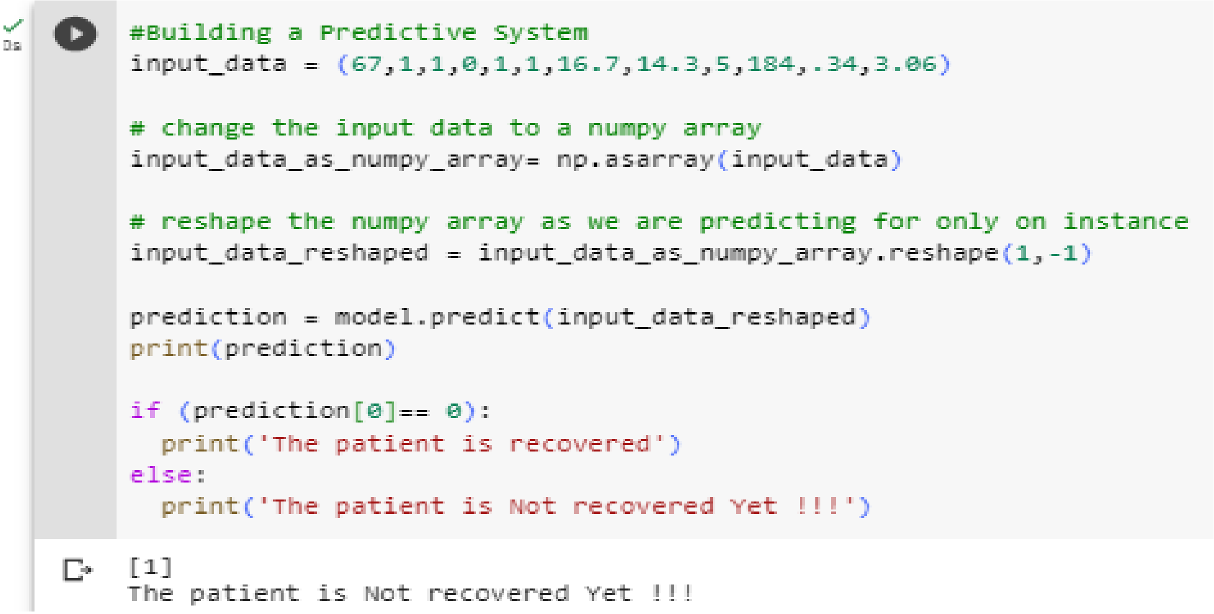
2.1.4.3 - Model Test: -

From a medical perspective:

Age = High	High probability of Covid
Neutroph’s count = High	High probability of Covid
Platelet’s count = Low	High probability of Covid
Monocytes’ count = Low	High probability of Covid
White blood count = Low	High probability of Covid
Antibiotic, paracetamol, oxygen Ventilation, red blood cell, gender	Contribute in diagnosis



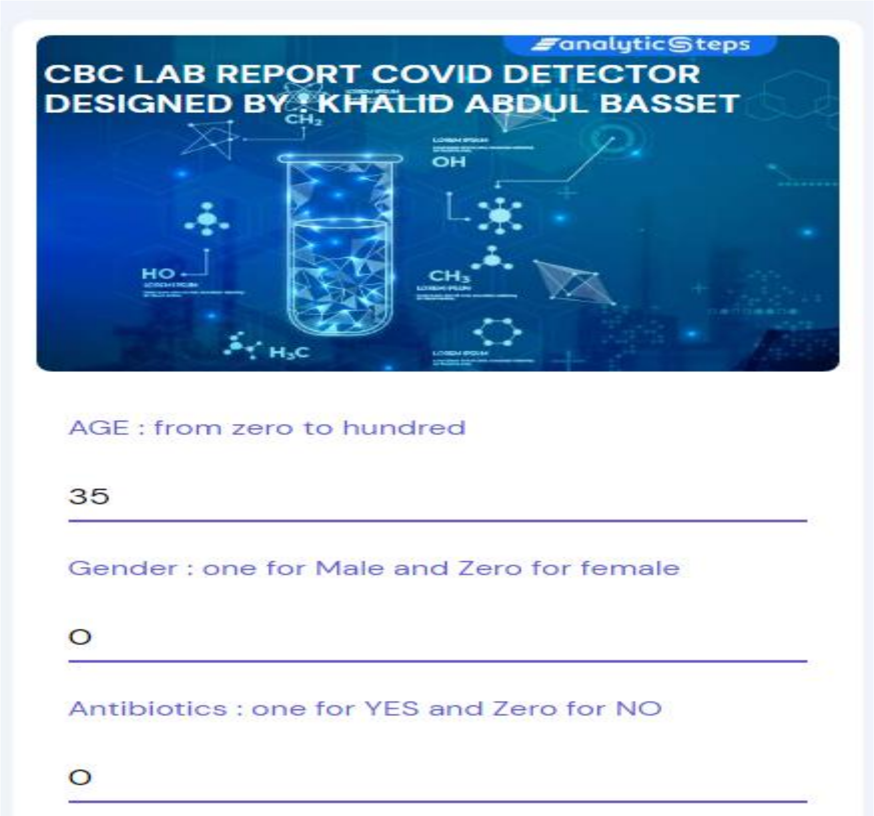
Patient Ag	Gender	Antibiotic	Paracetan	Oxcegen	Ventilate	Red blood	Monocyte	White blo	Platelet C	Lymphocy	Neutroph	Outcome
67	1	1	0	1	1	16.7	14.3	5	184	0.34	3.86	1



```
#Building a Predictive System
input_data = (67,1,1,0,1,1,16.7,14.3,5,184,.34,3.06)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The patient is recovered')
else:
  print('The patient is Not recovered Yet !!!')
```

```
[1]
The patient is Not recovered Yet !!!
```

2.2- Front-End Design of Web Application



CBC LAB REPORT COVID DETECTOR
DESIGNED BY : KHALID ABDUL BASSET

AGE : from zero to hundred

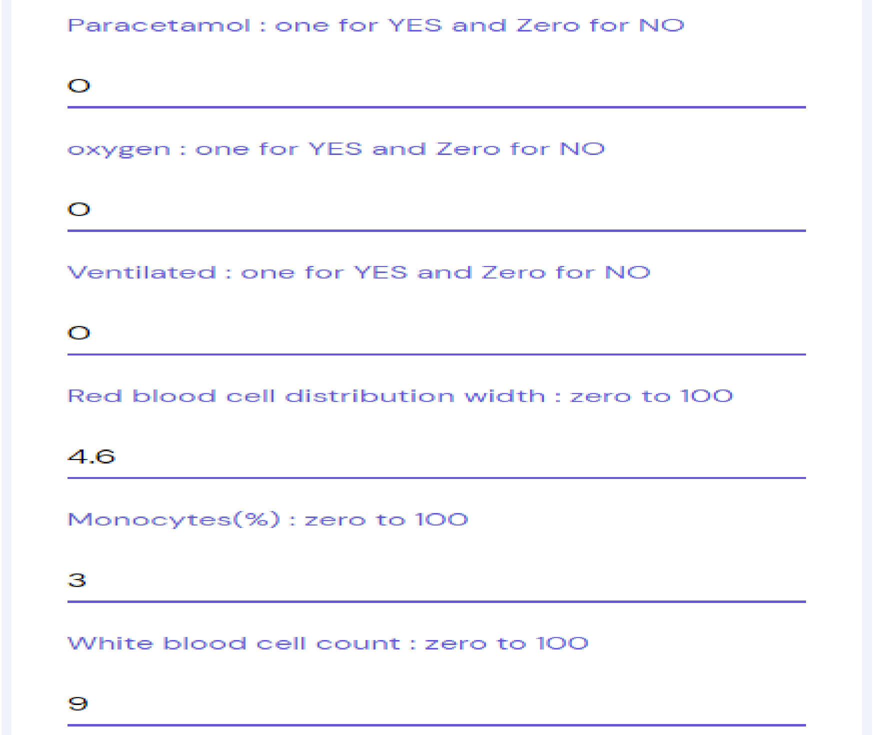
35

Gender : one for Male and Zero for female

0

Antibiotics : one for YES and Zero for NO

0



Paracetamol : one for YES and Zero for NO

0

oxygen : one for YES and Zero for NO

0

Ventilated : one for YES and Zero for NO

0

Red blood cell distribution width : zero to 100

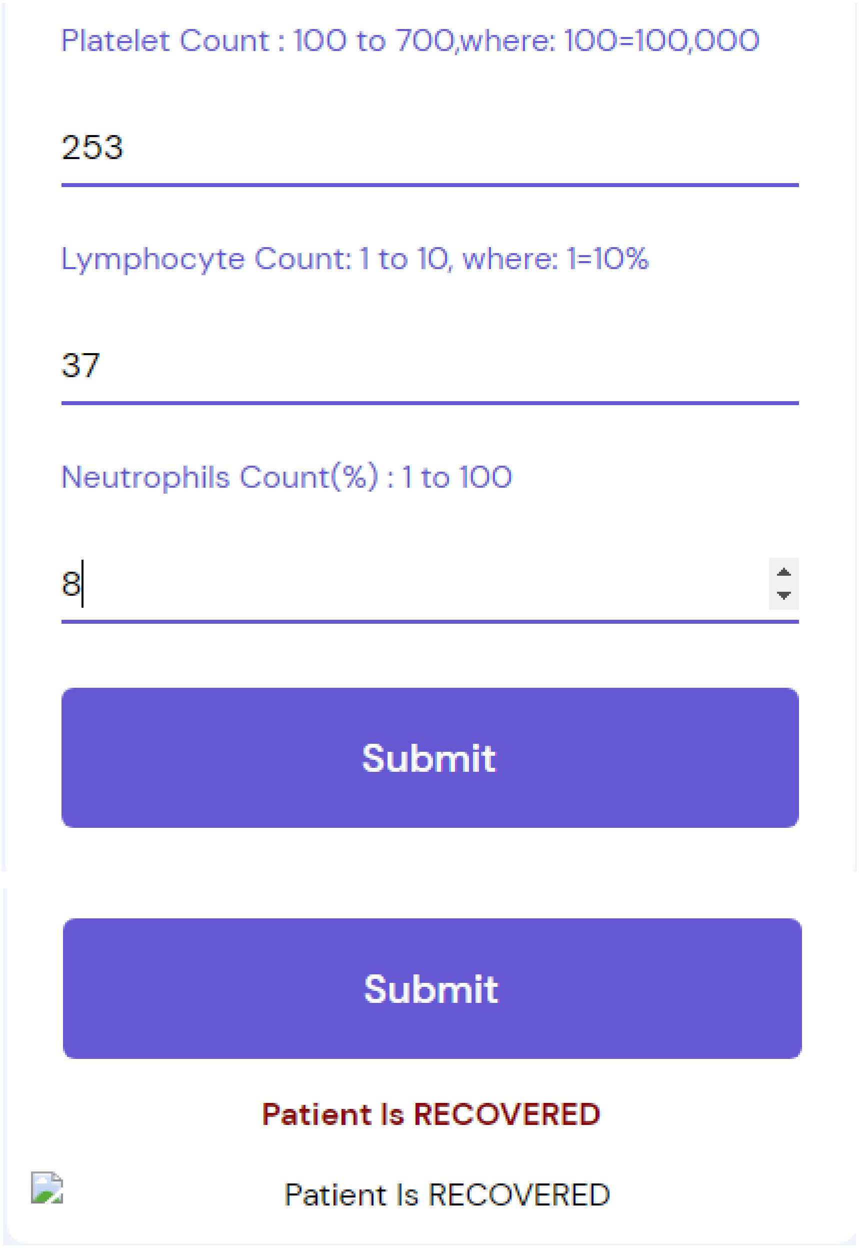
4.6

Monocytes(%) : zero to 100

3

White blood cell count : zero to 100

9



Platelet Count : 100 to 700,where: 100=100,000

253

Lymphocyte Count: 1 to 10, where: 1=10%

37

Neutrophils Count(%) : 1 to 100

8

Patient Is RECOVERED

Patient Is RECOVERED

Fig31: Screen shots of Lab report Detector (covid-19)

5- Implementation

Chapter Four: The Mobile Application

1- Flutter Introduction

Flutter is Google's portable UI framework for building modern, native, and reactive applications for Android, iOS, it's also an open-source project hosted on GitHub with contributions from Google and the community.

Flutter uses Dart, a modern object-oriented language that compiles to native ARM code and production-ready JavaScript code.

Dart is the language that you'll use to develop your Flutter applications, it is ahead-of-time (AOT) compiled to native code, making your Flutter application fast.

2- Why flutter framework

2.1- Fast Development

Flutter is faster than many other application development frameworks. With its "hot reload" feature, you can experiment, build UIs, add/remove features, test, and fix bugs faster.

Flutter's hot reload helps you see code and user interface changes immediately while retaining state to an app running the Dart virtual machine. In other words, every time you make code changes, you don't need to reload the app.

2.2- Expressive and Flexible UI

You can really build beautiful apps in Flutter. Also, the end-user experience is like native apps. Flutter has a layered architecture that lets you control every pixel on the screen. Thus, customization is very simple in Flutter. With its powerful composting capabilities, you can overlay and animate graphics, text, video, and other controls without any limitations. You'll also find a set of widgets that deliver pixel-perfect experiences on Android and iOS.

2.3- Native Performance

Flutter's widgets incorporate all critical platform differences such as scrolling, navigation, icons, and fonts. This provides native performance experience on both iOS and Android.

2.4- Dart Language

Dart programming language is developed by Google and is meant for mobile, desktop, backend, and web applications. It is a client-optimized language for fast performing apps on multiple platforms.

2.5- Important Tools

1. Widget inspector helps to visualize and explore the tree hierarchy. Flutter uses this for UI rendering.

2. Timeline view helps you to monitor your application at a frame-by-frame level, you can also identify rendering and computational work in

timeline view.

3. Source-level Debugger: It lets you step through code, set breakpoints, and investigate the call stack.

4. Logging View displays events from the Dart runtime, application frameworks and app-level logging events.

3- Flutter Widget

Flutter widgets are the building blocks of user interfaces in Flutter, a popular open-source mobile app development framework. Widgets are essentially reusable UI components that can be combined, customized for creating complex and dynamic user interfaces.

Flutter widgets are divided into two main categories: stateless, stateful. Stateless widgets are those that do not have any mutable state and can be drawn purely based on their configuration. Examples of stateless widgets include Text, Image, and Icon. Stateful widgets, on the other hand, have mutable state that can change over time, and they are responsible for managing their own state and updating their UI when needed.

Flutter widgets are highly customizable, and developers can using variety of properties and methods to change the appearance and behavior of a widget. For example, they can change the color, font, size, or padding of a widget, or they can add event listeners to respond to user actions.

4- Widget lifecycle

In Flutter, widgets can be either stateless or stateful. The main difference between these two types of widgets is how they handle changes to their properties or state over time.

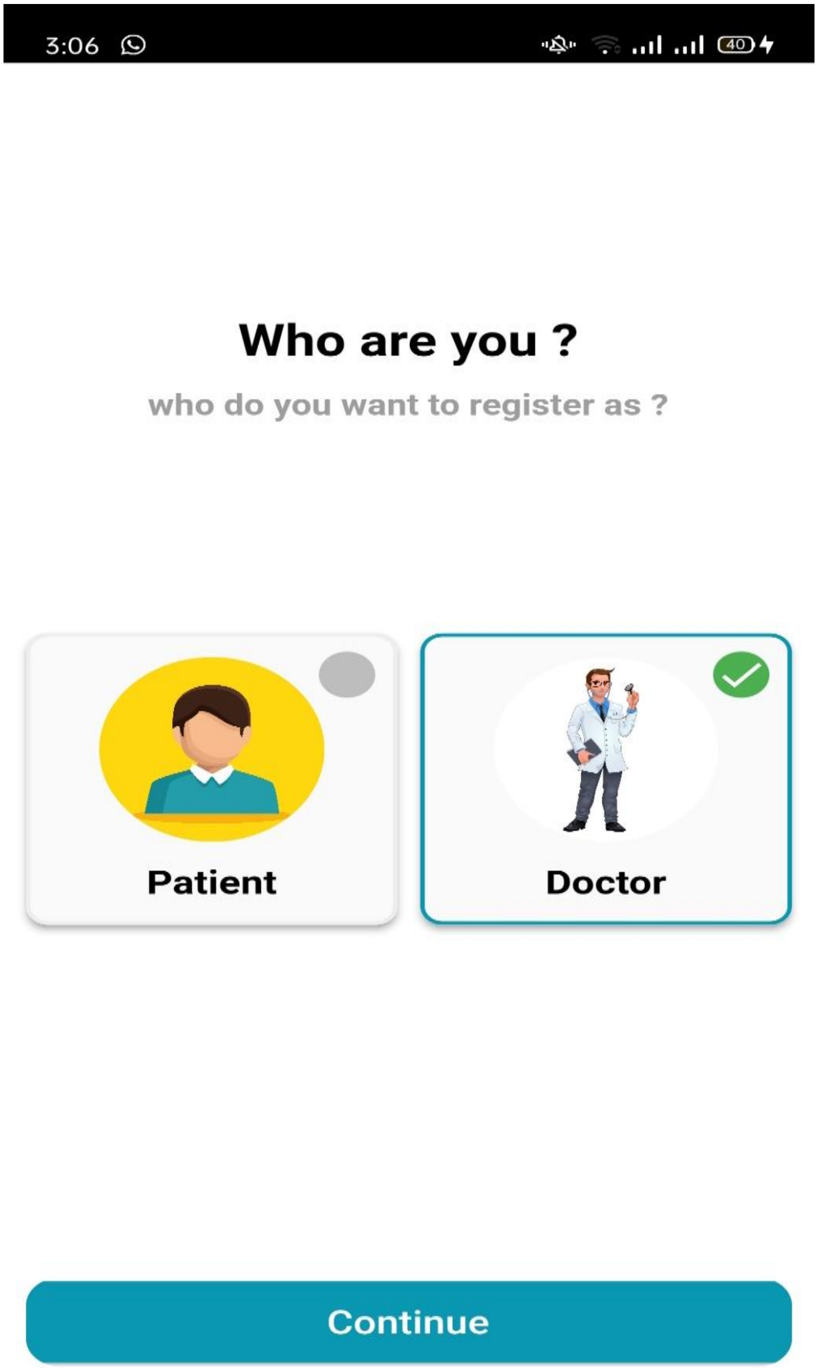
A stateless widget is one that does not have any mutable state. In other words, once a stateless widget is built, it cannot change its internal state. Instead, it simply takes a set of input properties and returns a widget tree that represents the user interface. For example, the Text widget is a stateless widget because it takes a string as input and returns a widget that displays that text.

A stateful widget, on the other hand, has mutable state that can change over time. Stateful widgets are used when you need to update the user interface in response to some user action or other event. When a stateful widget is built, it creates a State object that can hold mutable state. This State object can be updated over time, and when it changes, the widget rebuilds its user interface based on the new state. For example, a Checkbox widget is a stateful widget because it can change its internal state based on whether it is checked or unchecked.

To create a stateful widget in Flutter, you need to create two classes: one that extends Stateful Widget, and another that extends State.

6- Application (work flow and snap shots)

1. choose your state

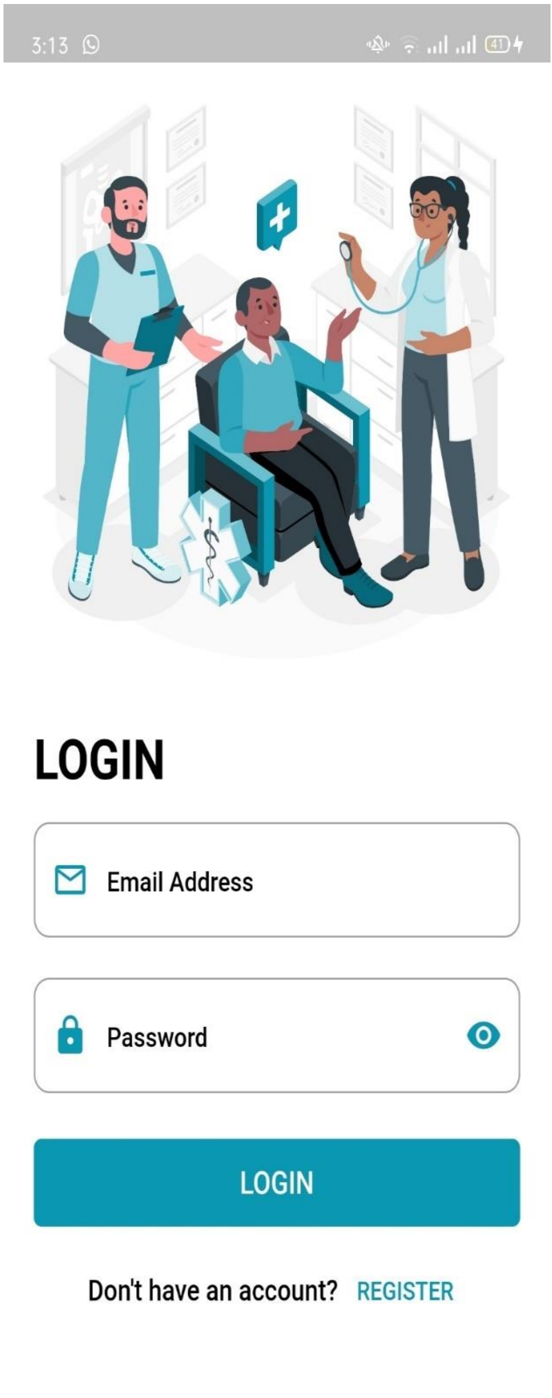


3:06

Who are you ?

who do you want to register as ?

2. Patient login and register



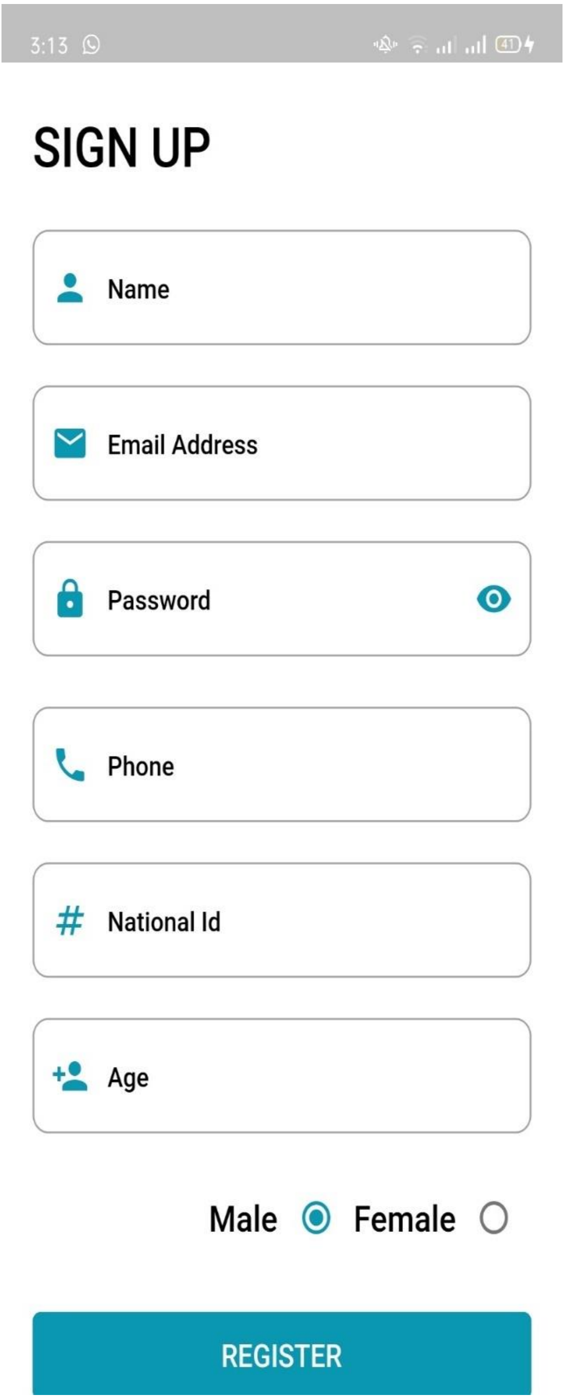
3:13

LOGIN

Email Address

Password

Don't have an account? REGISTER



3:13

SIGN UP

Name

Email Address

Password

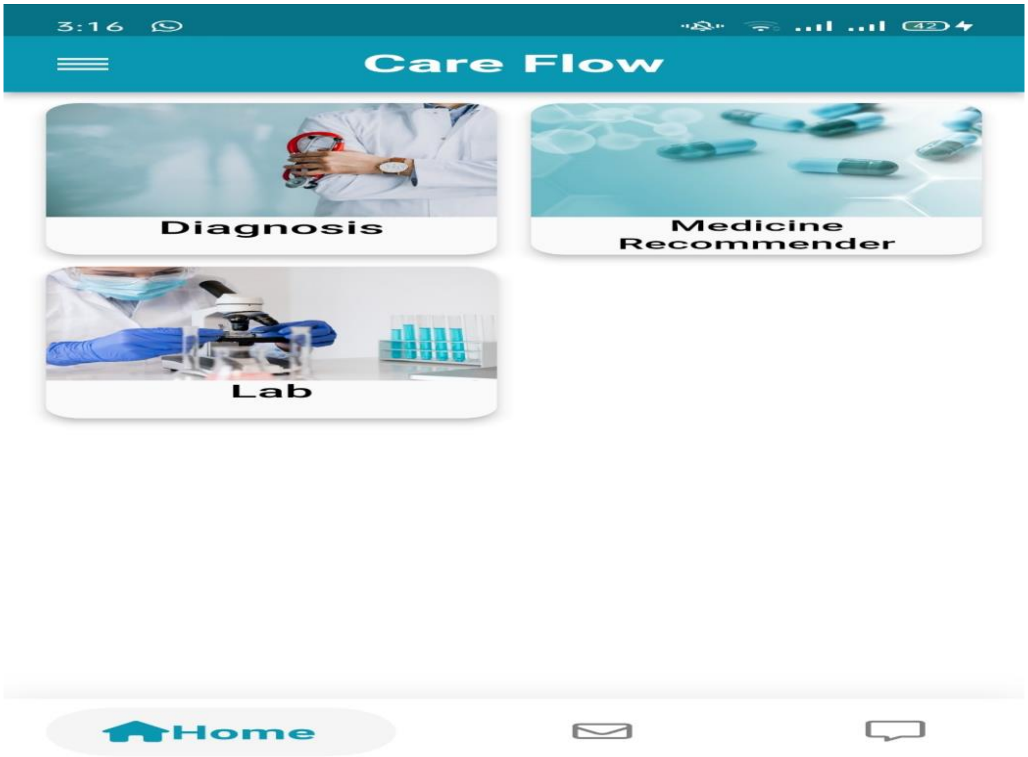
Phone

National Id

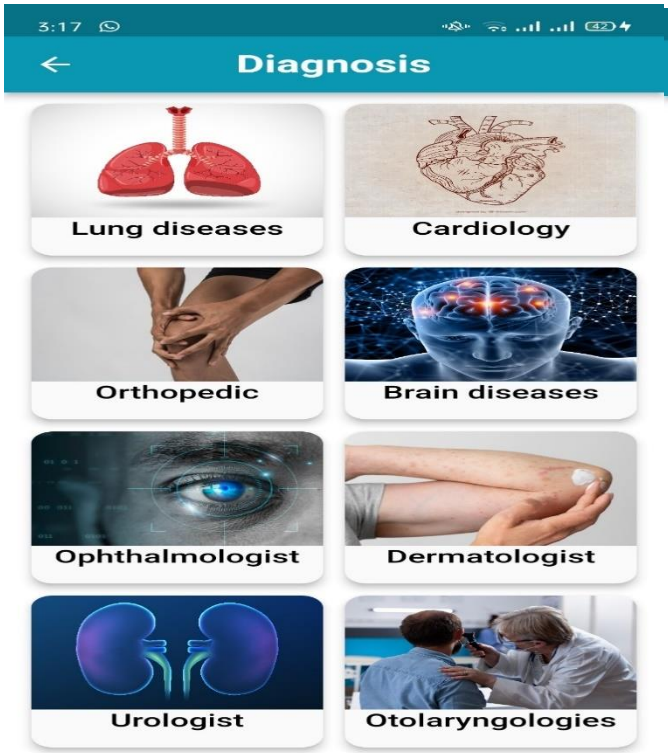
Age

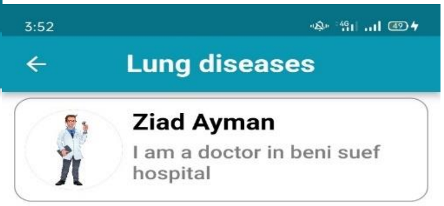
Male ● Female ○

3. Patient’s home page



4. choosing specialization

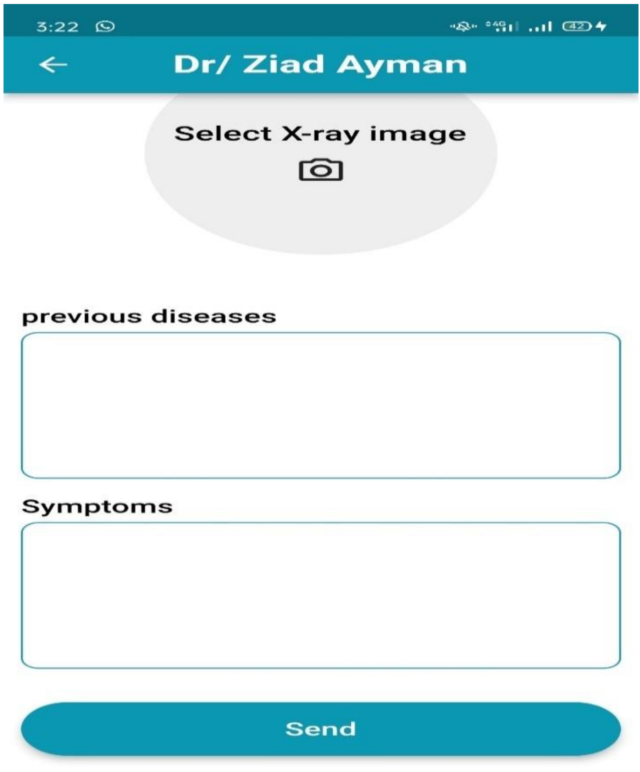


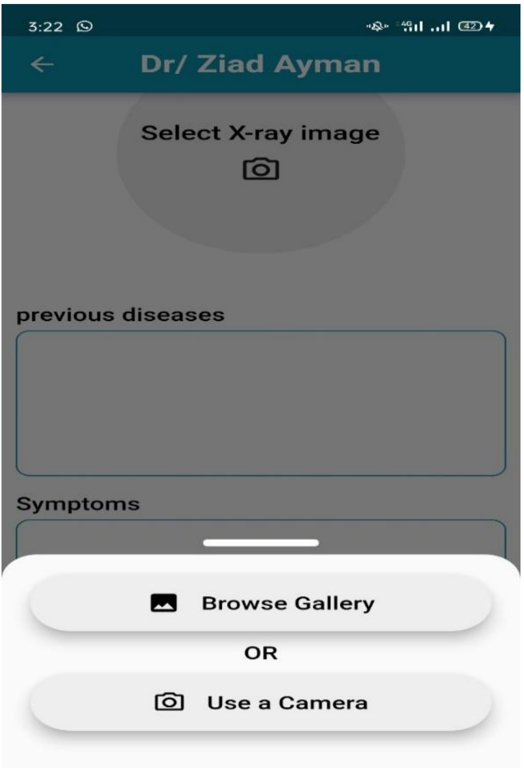


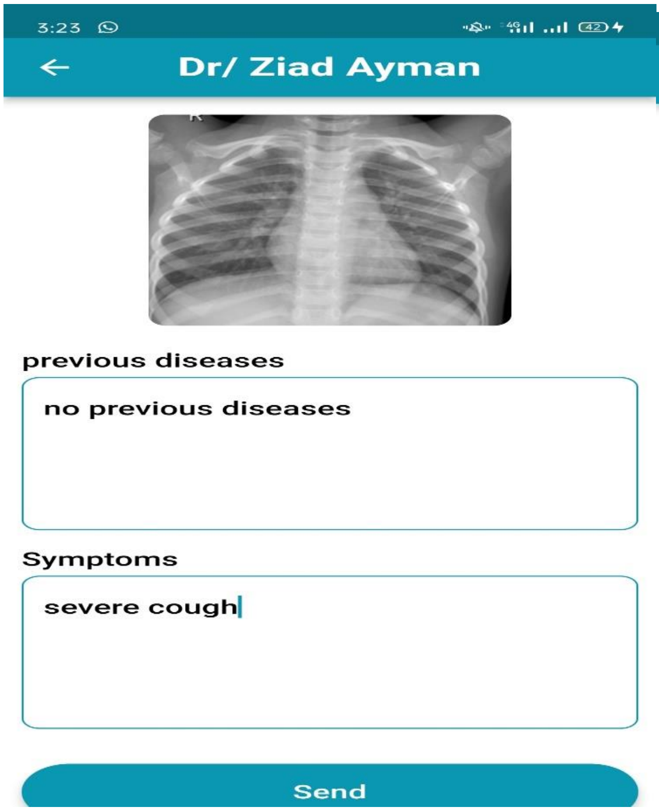
Ziad Ayman

I am a doctor in beni suef hospital

5. Sending request to be diagnosed by doctor

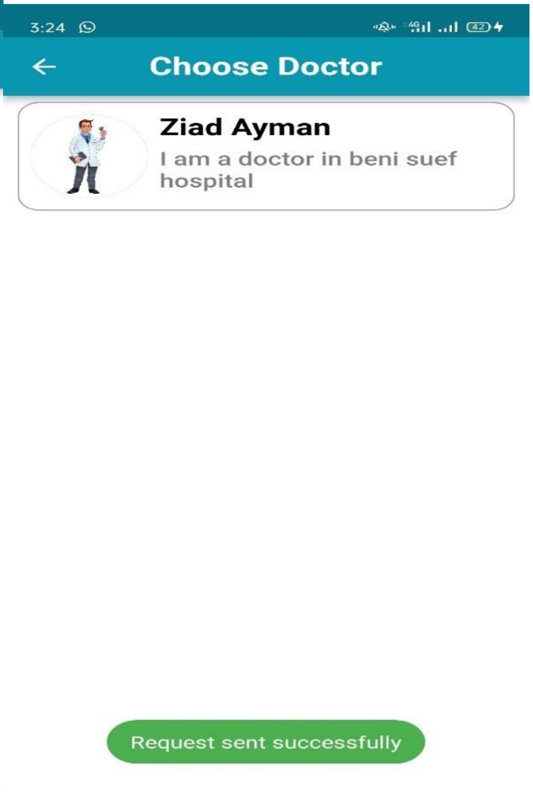






previous diseases: no previous diseases

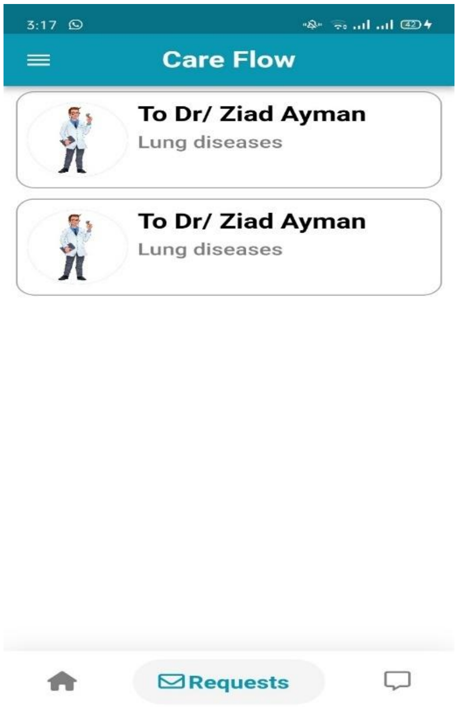
Symptoms: severe cough



Ziad Ayman
I am a doctor in beni suef hospital

Request sent successfully

6. The list of patient’s requests and doctor’s responses



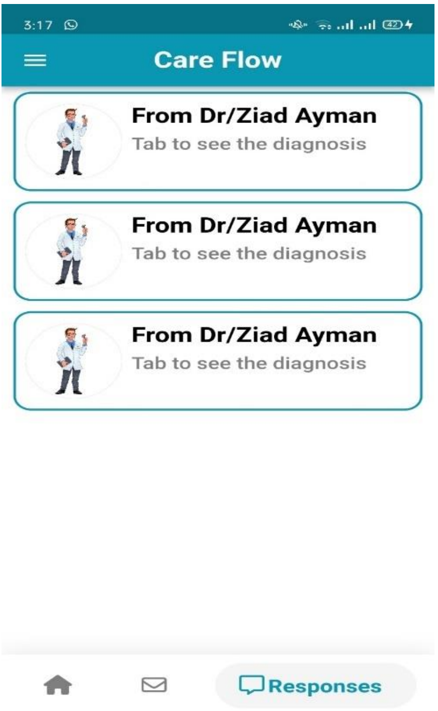
3:17

Care Flow

To Dr/ Ziad Ayman
Lung diseases

To Dr/ Ziad Ayman
Lung diseases

Requests



3:17

Care Flow

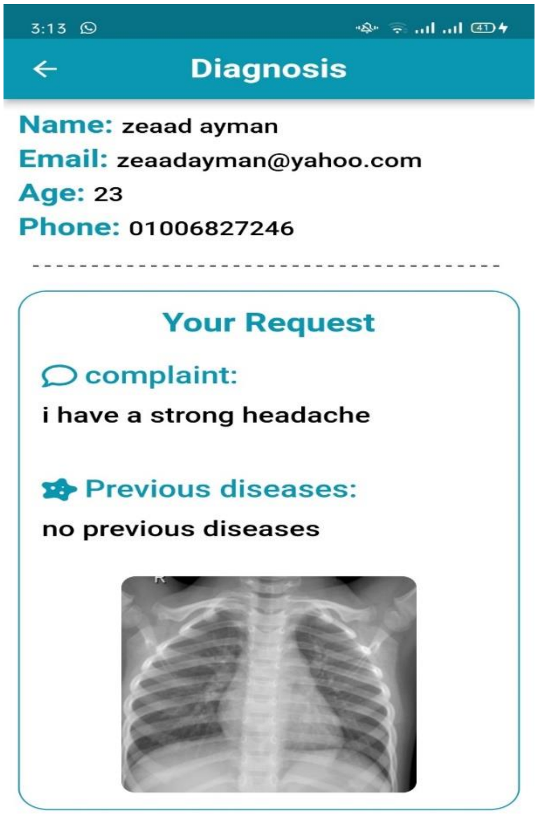
From Dr/Ziad Ayman
Tab to see the diagnosis

From Dr/Ziad Ayman
Tab to see the diagnosis

From Dr/Ziad Ayman
Tab to see the diagnosis

Responses

7. Doctor’s diagnosing with aid of Ai system



3:13

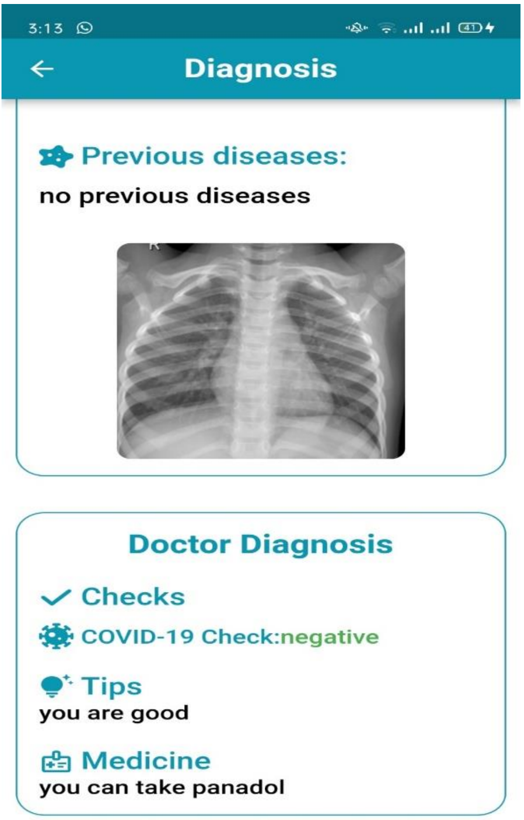
Diagnosis

Name: zeaad ayman
Email: zeaadayman@yahoo.com
Age: 23
Phone: 01006827246

Your Request

complaint:
i have a strong headache

Previous diseases:
no previous diseases



3:13

Diagnosis

Previous diseases:
no previous diseases

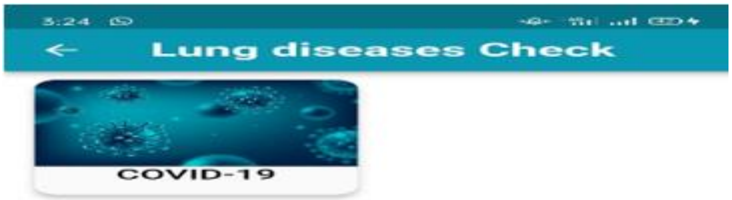
Doctor Diagnosis

✓ Checks
COVID-19 Check:negative

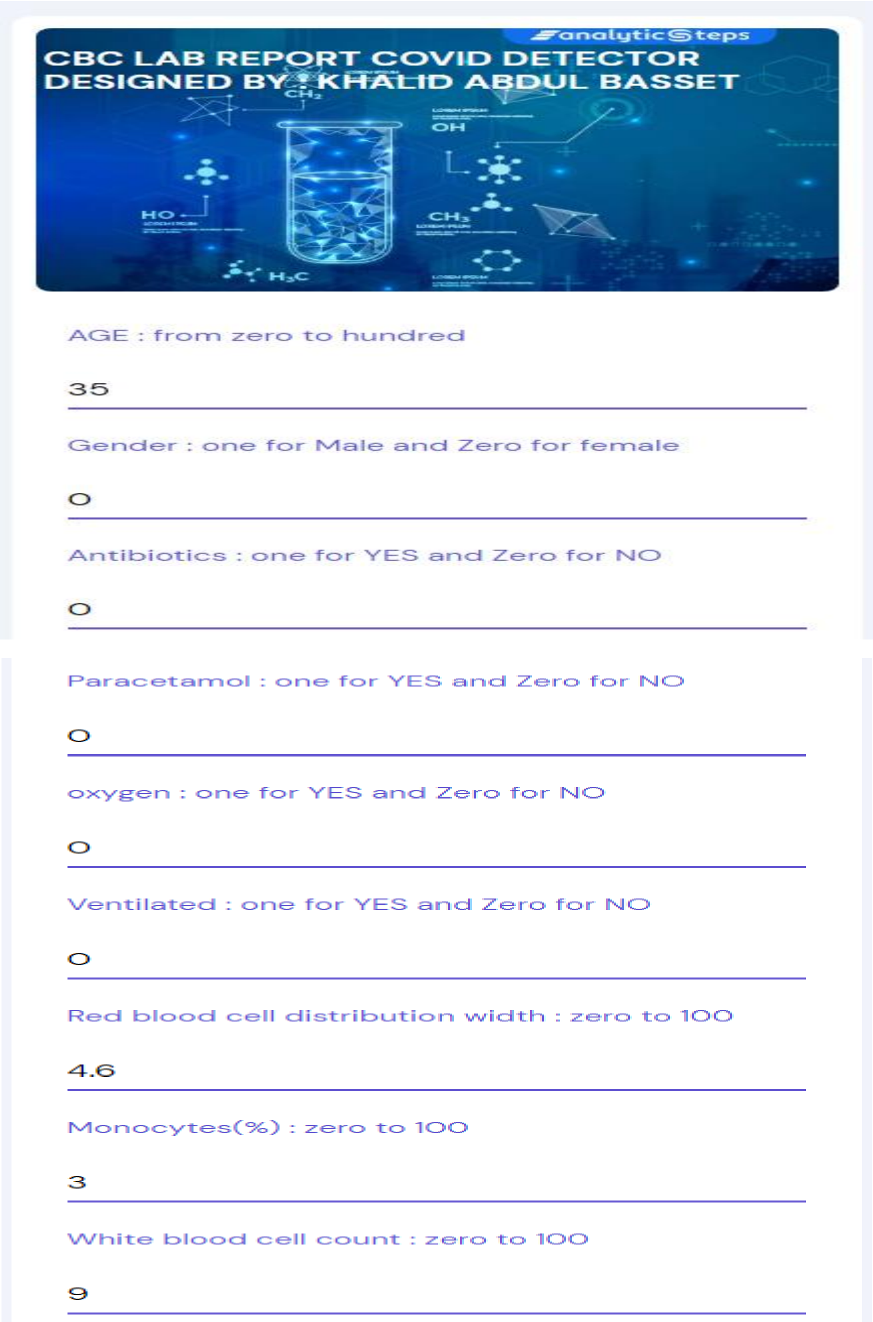
Tips
you are good

Medicine
you can take panadol

8. Doctor send diagnose to patients



9. patient tests his condition via Lab detector Checker



CBC LAB REPORT COVID DETECTOR
DESIGNED BY : KHALID ABDUL BASSET

AGE : from zero to hundred

35

Gender : one for Male and Zero for female

0

Antibiotics : one for YES and Zero for NO

0

Paracetamol : one for YES and Zero for NO

0

oxygen : one for YES and Zero for NO

0

Ventilated : one for YES and Zero for NO

0

Red blood cell distribution width : zero to 100

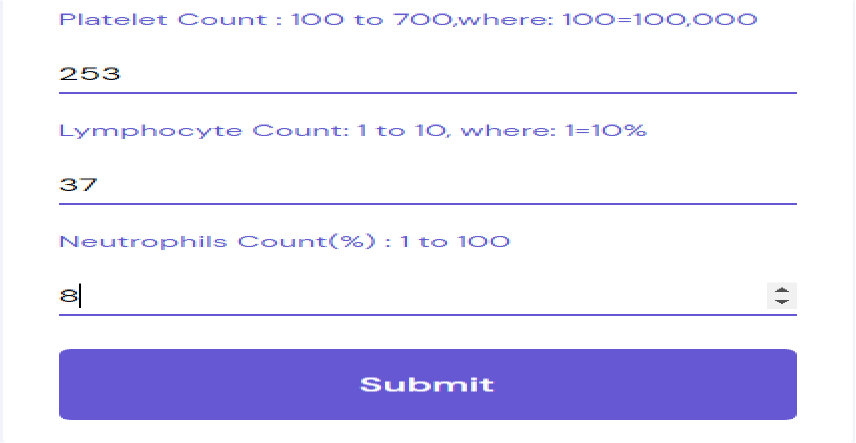
4.6

Monocytes(%) : zero to 100

3

White blood cell count : zero to 100

9



Platelet Count : 100 to 700,where: 100=100,000

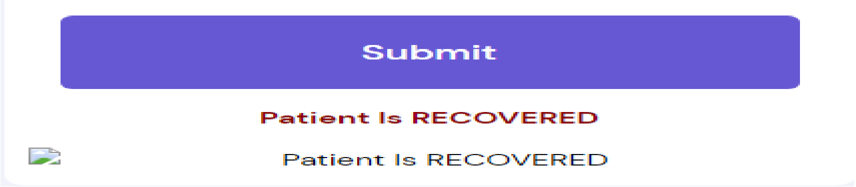
253

Lymphocyte Count: 1 to 10, where: 1=10%

37

Neutrophils Count(%) : 1 to 100

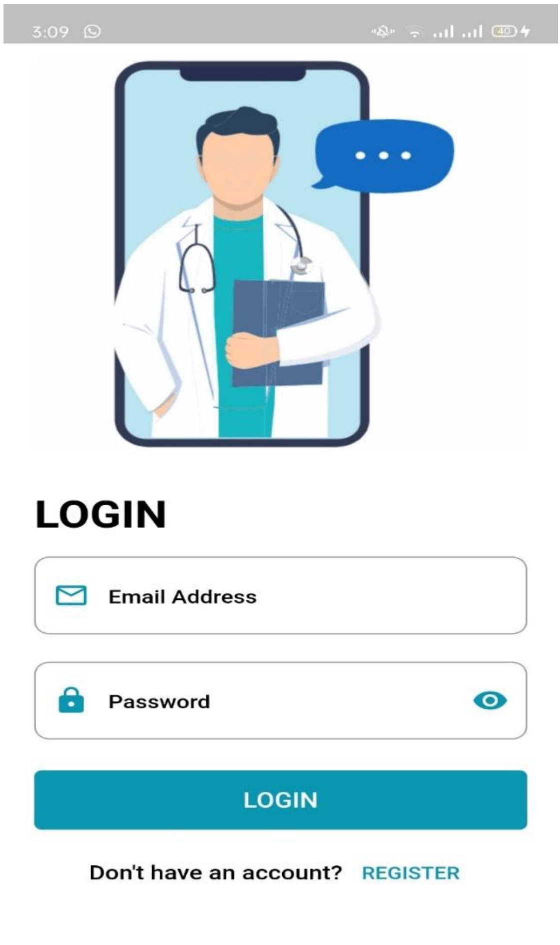
8



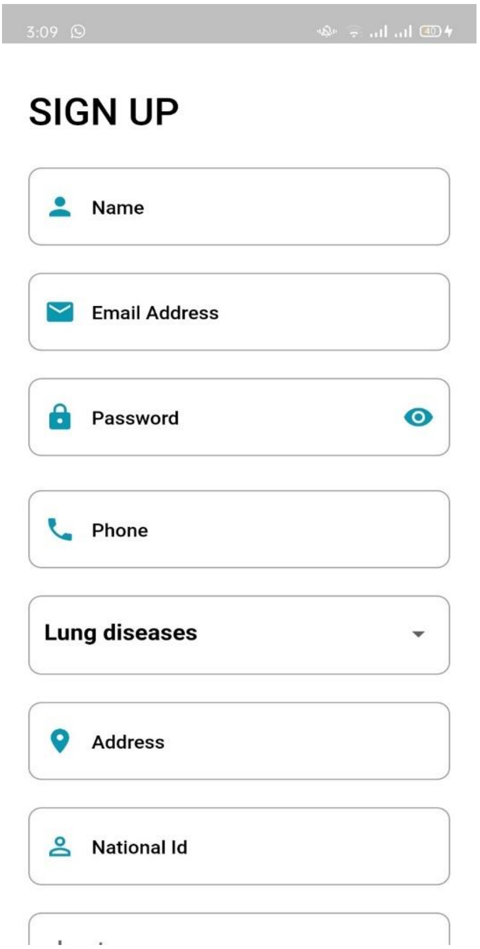
Patient Is RECOVERED

Patient Is RECOVERED

10. Doctor login and register



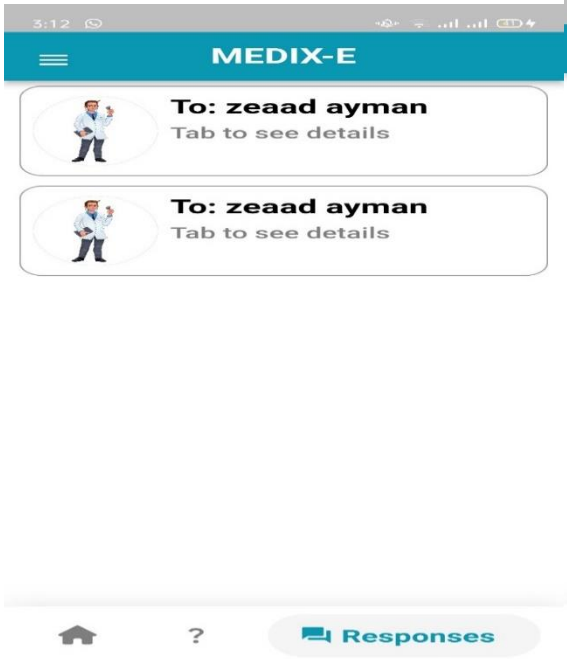
LOGIN



SIGN UP

Lung diseases

11. Received requests and sent diagnosis



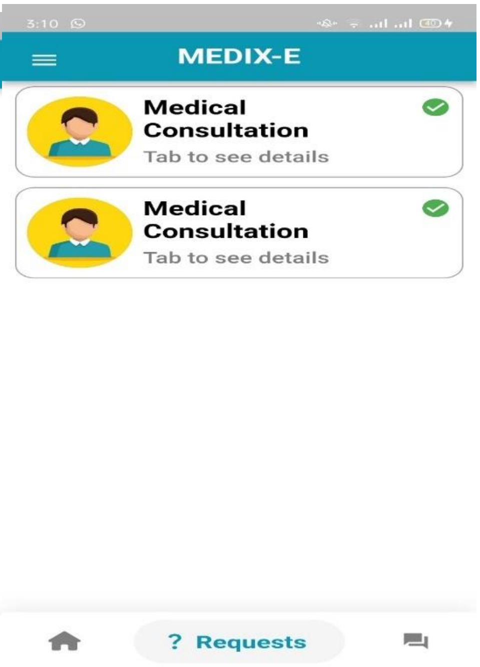
3:12

MEDIX-E

To: zeaad ayman
Tab to see details

To: zeaad ayman
Tab to see details

Responses



3:10

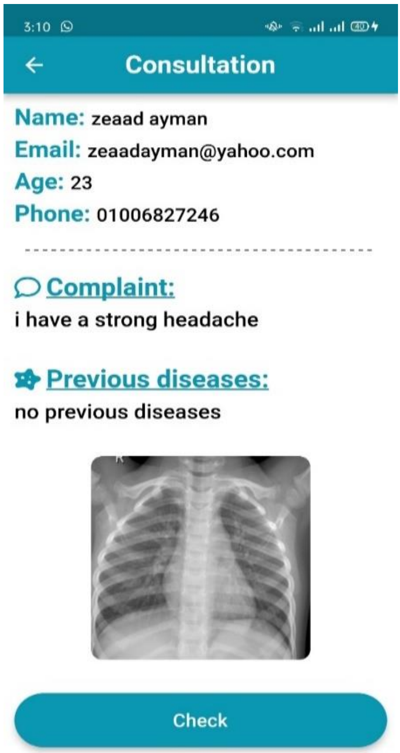
MEDIX-E

Medical Consultation ✓
Tab to see details

Medical Consultation ✓
Tab to see details

? Requests

12. Doctor open the request from patient and diagnosis



3:10

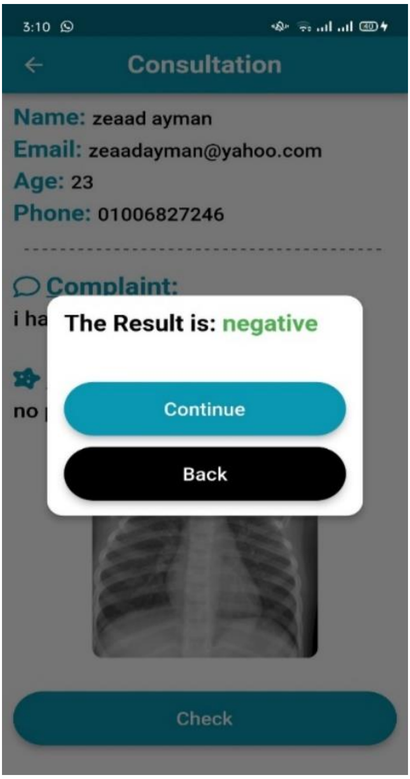
Consultation

Name: zeaad ayman
Email: zeaadayman@yahoo.com
Age: 23
Phone: 01006827246

Complaint:
i have a strong headache

Previous diseases:
no previous diseases

Check



3:10

Consultation

Name: zeaad ayman
Email: zeaadayman@yahoo.com
Age: 23
Phone: 01006827246

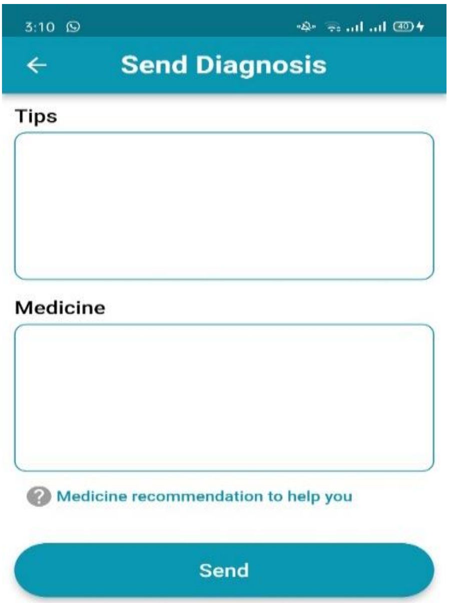
Complaint:

The Result is: negative

Continue

Back

Check



3:10

Send Diagnosis

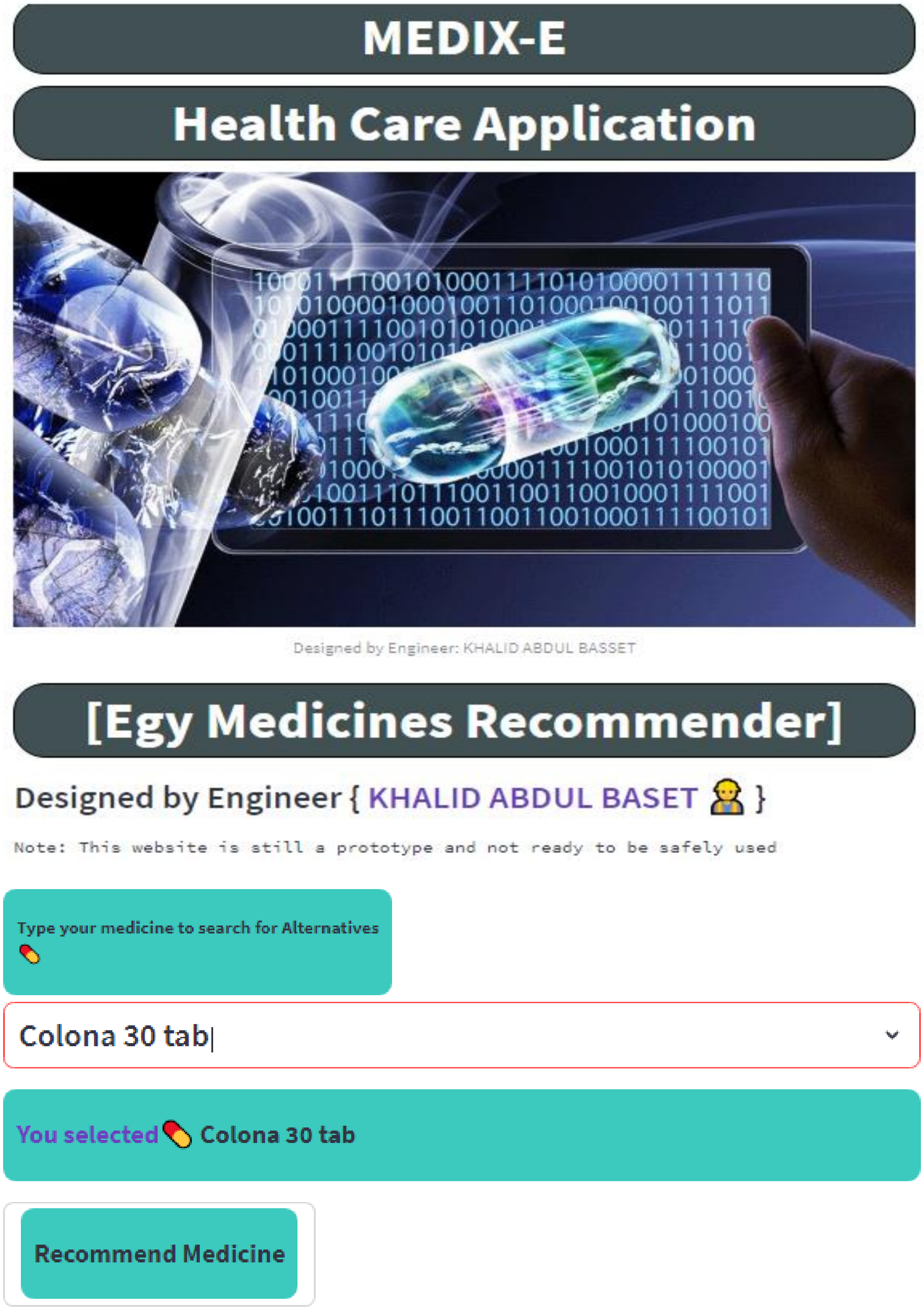
Tips

Medicine

Medicine recommendation to help you

Send

13. Doctor uses Medicine recommender



MEDIX-E

Health Care Application

Designed by Engineer: KHALID ABDUL BASSET

[Egy Medicines Recommender]

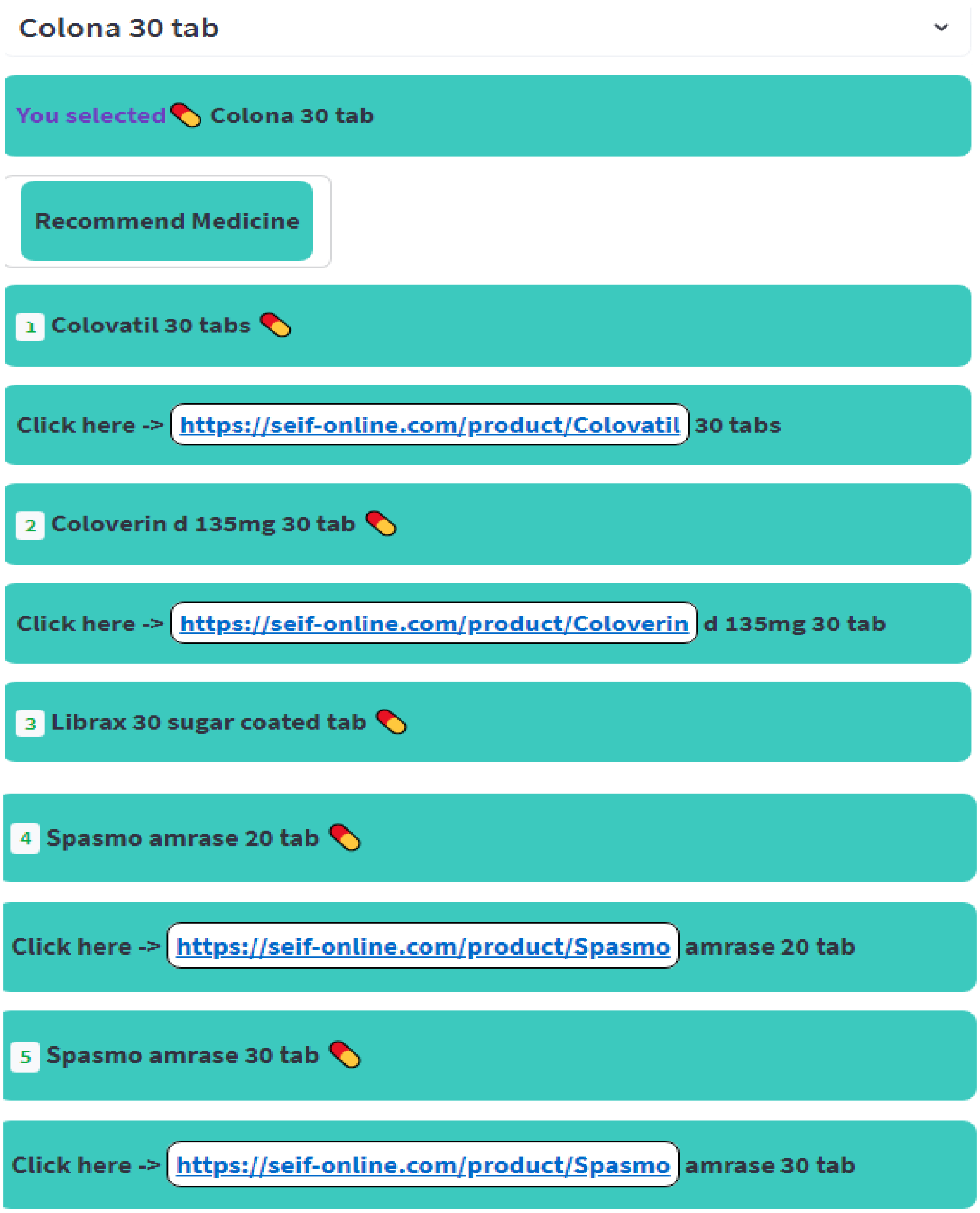
Designed by Engineer { KHALID ABDUL BASET 👷 }

Note: This website is still a prototype and not ready to be safely used

Type your medicine to search for Alternatives 💊

Colona 30 tab

You selected 💊 Colona 30 tab



Colona 30 tab ⌄

You selected 💊 Colona 30 tab

Recommend Medicine

1 Colovatil 30 tabs 💊

Click here -> https://seif-online.com/product/Colovatil 30 tabs

2 Coloverin d 135mg 30 tab 💊

Click here -> https://seif-online.com/product/Coloverin d 135mg 30 tab

3 Librax 30 sugar coated tab 💊

4 Spasmo amrase 20 tab 💊

Click here -> https://seif-online.com/product/Spasmo amrase 20 tab

5 Spasmo amrase 30 tab 💊

Click here -> https://seif-online.com/product/Spasmo amrase 30 tab

Recommended Medicines

7- Firebase

Firebase is a mobile and web application development platform that provides a variety of backend services to help developers build high-quality apps quickly and easily.

Firebase provides a wide range of services and features, including:

1- Realtime Database: A cloud-hosted NoSQL database that allows developers to store and sync data in real-time.

2- Authentication: A service that allows developers to add user authentication to their app, including email and password authentication, social login, and more.

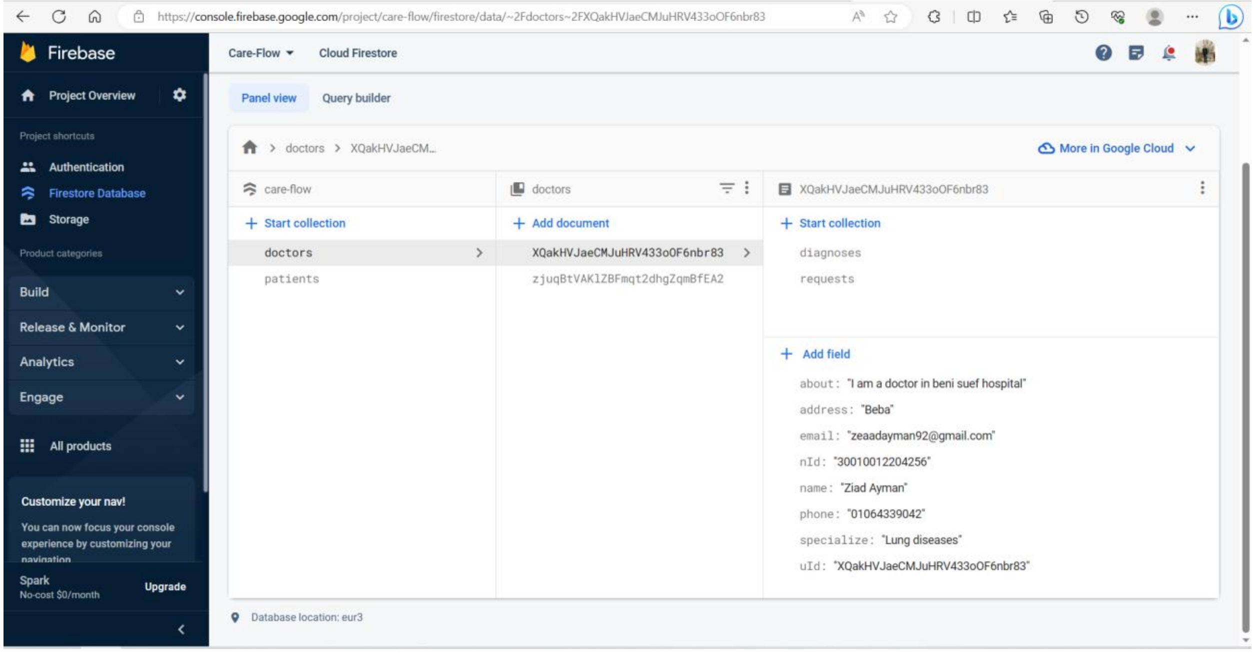
3- Firebase firestore: A cloud-hosted NoSQL database that allows developers to store data in collections and documents. It is not real-time.

4- Cloud Functions: A serverless platform that enables developers to run server-side code in response to events.

5- Cloud Storage: A service that allows developers to store and serve user-generated content such as images, videos, and audio files.

6- Cloud Messaging: A service to send targeted push notifications.

Snapshots from our database



care-flow	doctors	XQakHVJaeCMJuHRV433oOF6nbr83
doctors	XQakHVJaeCMJuHRV433oOF6nbr83	diagnoses
patients	zjuqBtVAKlZBFmqt2dhgZqmBfEA2	requests

about: "I am a doctor in beni suef hospital"

address: "Beba"

email: "zeaadayman92@gmail.com"

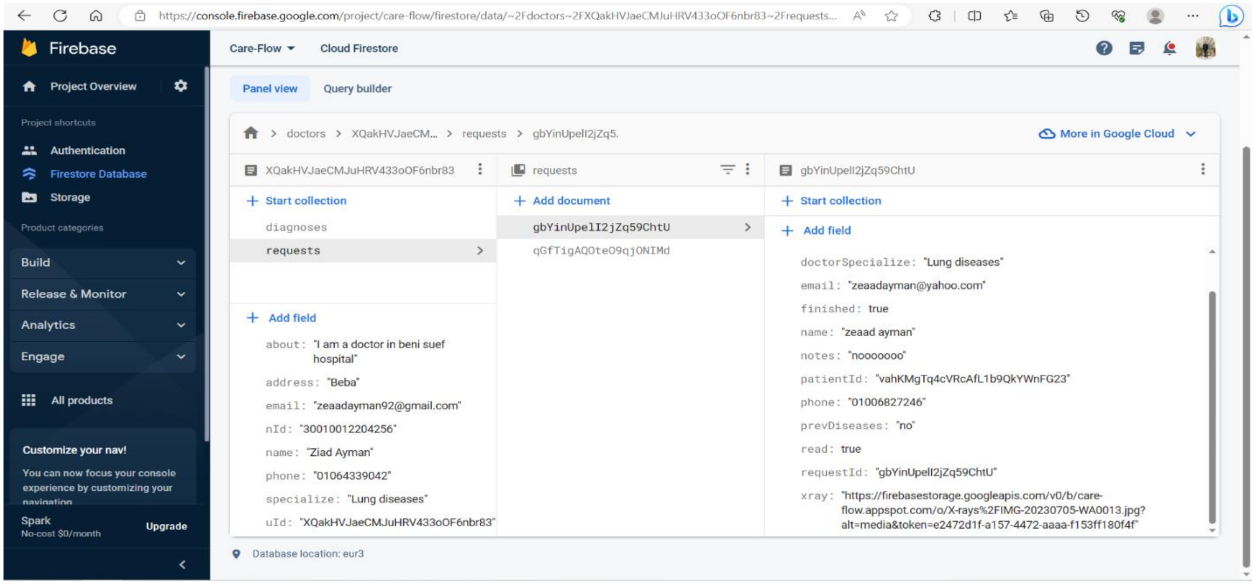
nId: "30010012204256"

name: "Ziad Ayman"

phone: "01064339042"

specialize: "Lung diseases"

uId: "XQakHVJaeCMJuHRV433oOF6nbr83"



XQakHVJaeCMJuHRV433oOF6nbr83	requests
diagnoses	gbYinUpelI2jZq59ChtU
requests	qGfTigAQOteO9qjONIMd

about: "I am a doctor in beni suef hospital"

address: "Beba"

email: "zeaadayman92@gmail.com"

nId: "30010012204256"

name: "Ziad Ayman"

phone: "01064339042"

specialize: "Lung diseases"

uId: "XQakHVJaeCMJuHRV433oOF6nbr83"

gbYinUpelI2jZq59ChtU

doctorSpecialize: "Lung diseases"

email: "zeaadayman@yahoo.com"

finished: true

name: "zeaad ayman"

notes: "noooooooo"

patientId: "vahKMgTq4cVRcAfL1b9QkYWnFG23"

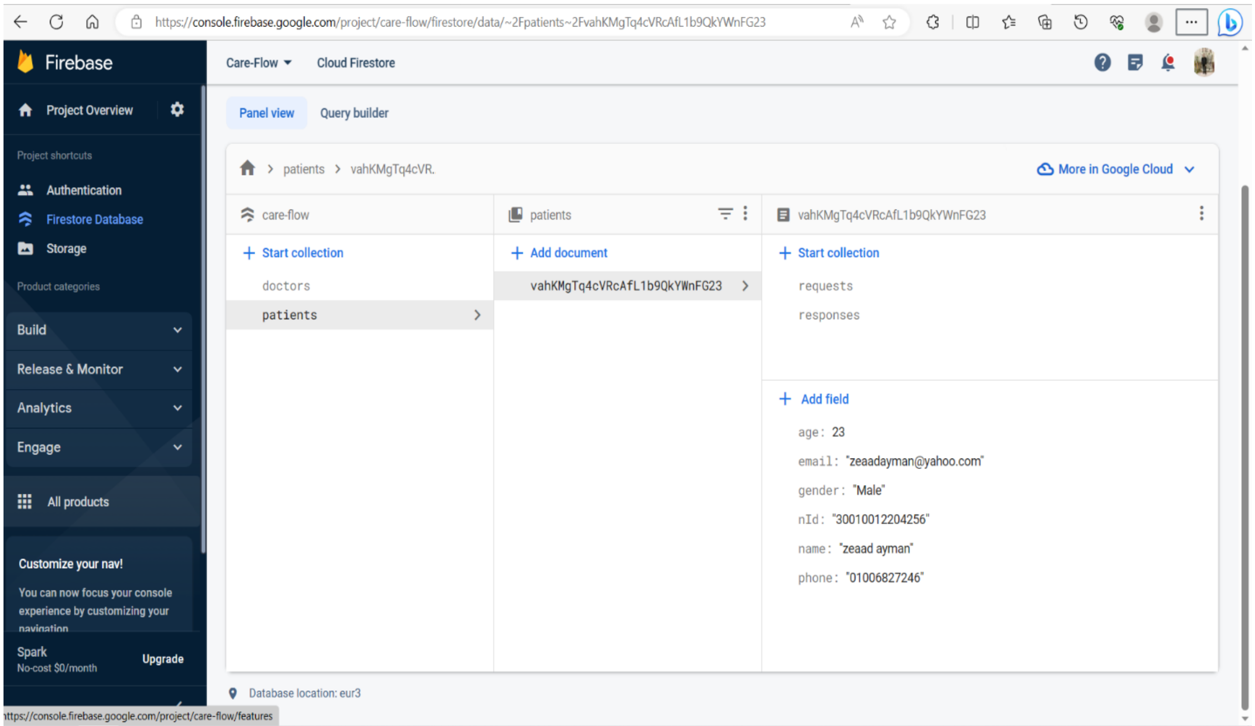
phone: "01006827246"

prevDiseases: "no"

read: true

requestId: "gbYinUpelI2jZq59ChtU"

xray: "https://firebasestorage.googleapis.com/v0/b/care-flow.appspot.com/o/X-rays%2FIMG-20230705-WA0013.jpg?alt=media&token=e2472d1f-a157-4472-aaaa-f153ff180f4f"

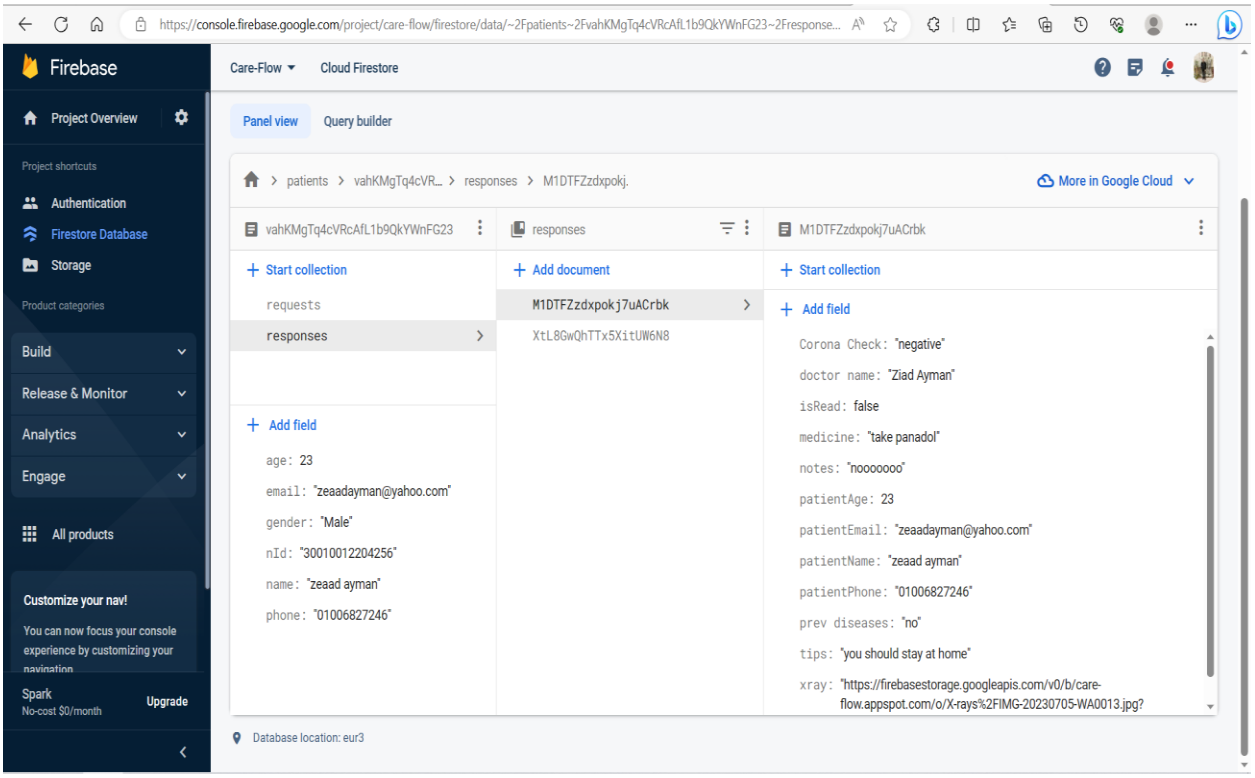


doctors
patients

vahKMgTq4cVRcAfL1b9QkYWnFG23

requests
responses

age: 23
email: "zeaadayman@yahoo.com"
gender: "Male"
nId: "30010012204256"
name: "zeaad ayman"
phone: "01006827246"



requests
responses

age: 23
email: "zeaadayman@yahoo.com"
gender: "Male"
nId: "30010012204256"
name: "zeaad ayman"
phone: "01006827246"

M1DTFZzdxpokj7uACrbk
XtL8GwQhTTx5XitUW6N8

Corona Check: "negative"
doctor name: "Ziad Ayman"
isRead: false
medicine: "take panadol"
notes: "nooooooo"
patientAge: 23
patientEmail: "zeaadayman@yahoo.com"
patientName: "zeaad ayman"
patientPhone: "01006827246"
prev diseases: "no"
tips: "you should stay at home"
xray: "https://firebasestorage.googleapis.com/v0/b/care-flow.appspot.com/o/X-rays%2FIMG-20230705-WA0013.jpg?

8- Testing

1.Sign up as a doctor

Test case id	Test Objective	Precondition	Steps	Test data	Expected Result	Actual Result
TC_01	Successful sign up	Valid email, Password, phone, name, Specialization, NId	1. enter name 2.enter email 3.enter password 4.enter phone 5.enter NId	Name=doctor Email=doctor@gmail.com password= 12345678 phone= 01064339042 NId =30010012205298	Sign up successfully and go to login page	Sign up successfully and go to login page
TC_02	Make sure that a message will appear to tell you that the NId or phone are invalid	Valid email and password. Invalid phone and NId	1. enter name 2.enter email 3.enter password 4.enter phone 5.enter NId	Name=doctor Email=doctor@gmail.com Password=12345678 Phone=0106433904 NId=3001001220529	Message will appear to tell the doctor to enter valid NId	The message appeared
TC_03	Make sure that a message will appear to tell you that the email is invalid	Invalid email, Valid password, Valid phone, Valid NId	1. enter name 2.enter email 3.enter password 4.enter phone 5.enter NId	Name=doctor Email=doctor@ password=12345678 phone=01064339042 NId=30010012205298	Message will appear to tell the doctor to enter valid email	The message appeared
TC_04	Make sure that a message will appear to tell you that the field not empty	Do not enter any field	Do not enter any field		Message will appear to tell the doctor that the field cannot be empty	The message appeared

2. Login as a doctor

TC_05	Login successfully	Valid email and password.	1. enter the email 2.enter the password 3. click on login	Email=doctor@gmail.com Password=12345678	Login successfully and go to home	Login successfully and go to home
TC_0	Make sure that a message appears to tell the doctor that the email or password are wrong	Invalid email or password	1. enter valid email 2.enter invalid password	Email=doctor@gmail.com Password=1234	Message will appear to tell the user that the email or password are wrong	The message appears to tell the user that the email or password are wrong

3. The Patient sends a request to the doctor

TC_07	The request sent successfully	Enter all requirements	1. enter the previous diseases 2. enter the symptoms 3. upload the X-ray image	Prev diseases= heart disease Symptoms= Severe cough Upload the image	The request will be sent successfully	The request has been sent successfully
TC_08	Make sure that a message appears to tell the patient to enter the symptoms	Do not enter the symptoms	1. enter the previous diseases 2. upload the X-ray image	Prev diseases= heart disease Upload the image	Message will appear to tell the user to write the symptoms	The message has been appeared to the user
TC_09	Make sure that a message to upload xray	Do not upload the image	1. enter diseases. the Symp	diseases= heart , Symp= cough	Message will appear to upload image	The message has been appeared to the user

4. the doctor sends a response to the patient

TC_10	The response sent successfully	Enter all requirements	1. write the diagnosis 2. write the recommended drugs	Diagnosis= You have COVID19 and you should stay at home. Drugs=you can take Panadol	The response will be sent successfully	The response has been sent successfully
TC_11	Make sure that a message appears to tell the doctor to write the diagnosis	Do not enter the diagnosis	1. write the recommended drugs	Drugs=you can take Panadol	Message will appear to the doctor to tell him to write the diagnosis	The message has been appeared

<u>9- References</u>

- NTI National Technology Institute’s Huawei machine learning diploma

- An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification.

  - P:Aijaz Ahmad Reshi, Furqan Rustam, Arif Mehmood, Abdulaziz Alhossan.

- COVID-19 Detection using Convolutional Neural Network Architectures based upon Chest X-rays Images.

  - Publisher: IEEE

- Recommender Systems: An Overview
    - Authors: Robin Burke, Alexander Felfernig, Mehmet H. Göker.

- Recommendation systems: Principles, methods and evaluation.
    - Authors: Isinkaye a Folajimi , Ojokoh.

- Medicine recommender system: A machine learning approach December 2022
    - Authors: Amit Abhishek, Dharminder Kumar Bindal
- QCovSML: A reliable COVID-19 detection system using CBC biomarkers by a stacking machine learning model.
    - Authors: Tawsifur Rahman a, Amith Khandakar a, Farhan Fuad Abir,….

- Bacteremia detection from complete blood count and differential leukocyte count with machine learning: complementary and competitive with C-reactive protein and procalcitonin tests
    - Authors: Frank Lien, Huang-Shen Lin
- ITI Information Technology Institution android application developing diploma
- Android Documentation
- Flutter documentation