MediSafe – Stay away and defeat diseases

2022 - 143

SLIIT

Discover Your Future

Our Team

Supervisor

Mr. Ravi Supunya

Co- Supervisor

Mr. Samantha Rajapaksha

External Supervisor

Dr. Shashika Liyanage

IT18077698

Thennakoon T.M.B.C.K

IT19015422

Perera B.A.A.W.S

IT19015040

Rasuni Wageesha H.A

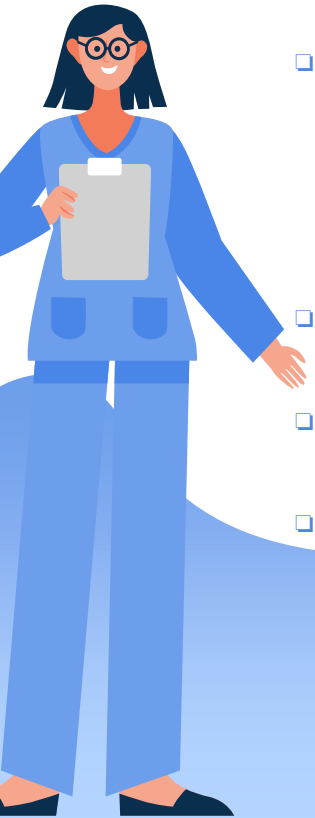
IT19011608

Senanayaka S.A.M.A.B.M

Introduction

- What is MediSafe ?
- Why are we doing this research?
- What is the purpose of our research?

Research problem



- There are some diseases that have arisen at present. (Heart attack, Pneumonia, Wheezing, Dengue, Covid'19)
- High cost for diagnosis.
- Informal lifestyle and busyness.
- Don't have enough idea about current situation of the country.

Background

Table 22. Leading Causes of Hospital Deaths, 2010 - 2019

Disease and ICD (10th Revision) Code		2019		2018		2017		2016		2015		2014		2013		2012		2011 [2]		2010 [2]	
		Rank	%	Rank	%	Rank	%	Rank	%	Rank	%	Rank	%	Rank	%	Rank	%	Rank	%	Rank	%
Ischaemic heart disease	I20 - I25	1	15.1	1	15.0	1	14.2	1	14.1	1	14.2	1	14.8	1	14.7	1	14.4	1	13.4	1	12.8
Zoonotic and other bacterial diseases	A20 - A49	2	12.1	3	10.9	2	11.5	3	11.6	3	9.7	3	9.1	6	7.9	6	7.1	6	6.7	6	6.6
Neoplasms[1]	C00 - D48	3	11.7	2	11.7	3	10.5	2	12.0	2	11.0	2	11.7	2	11.2	2	11.6	2	11.8	2	11.1
Diseases of the respiratory system excluding diseases of upper respiratory tract, pneumonia and influenza	J20 - J22, J40 - J98	4	10.7	4	9.9	4	9.8	5	8.3	4	9.2	6	8.0	5	7.9	5	7.2	5	6.9	5	7.0
Pneumonia	J12 - J18	5	8.0	7	7.8	6	8.2	7	6.4	7	7.5	7	6.6	8	6.1	8	5.7	9	5.2	9	5.2
Pulmonary heart disease and diseases of the pulmonary circulation	I26 - I51	6	7.6	6	7.9	5	8.5	4	8.7	5	8.3	4	8.6	4	8.4	3	9.0	4	8.7	3	8.7
Cerebrovascular disease	I60 - I69	7	7.6	5	8.0	7	7.7	6	8.2	6	8.2	5	8.4	3	8.6	4	8.7	3	8.7	4	8.7
Diseases of the urinary system	N00 - N39	8	5.8	8	5.8	8	5.9	8	6.3	8	6.2	8	6.3	7	6.2	7	6.3	7	5.7	8	5.7
Diseases of the gastro-intestinal tract	K20 - K92	9	5.0	9	5.1	9	5.1	9	5.5	9	5.3	9	5.7	9	5.7	9	5.4	8	5.4	7	6.2
Traumatic injuries	S00 - T39, W54	10	3.6	10	3.9	10	3.8	10	3.9	10	3.8	10	3.5	11	3.3	11	3.7	11	3.6	11	3.7
Disease of the nervous system	G00 - G98	11	1.3	13	1.4	14	1.4	14	1.4	17	1.3	16	1.4	15	1.4	16	1.5	19	1.4	18	1.6
Symptoms, signs and abnormal clinical and labo	R00 - R99	12	1.3	11	1.5	12	1.5	12	1.6	13	2.3	11	3.2	10	4.8	10	4.5	10	4.1	10	5.0
Diabetes mellitus	E10 - E14	13	1.3	12	1.4	11	1.7	11	1.6	13	1.6	13	1.6	13	1.6	14	1.7	14	1.9	16	1.7

[1] Includes deaths reported from the Cancer Hospital (not analysed by site and type of neoplasm)

[2] Excludes Mullaitivu District

Source : Medical Statistics Unit

Source : Medical Statistics Unit Ministry of Health
http://www.health.gov.lk/moh_final/english/public/elfinder/files/publications/AHB/AHS%202019.pdf

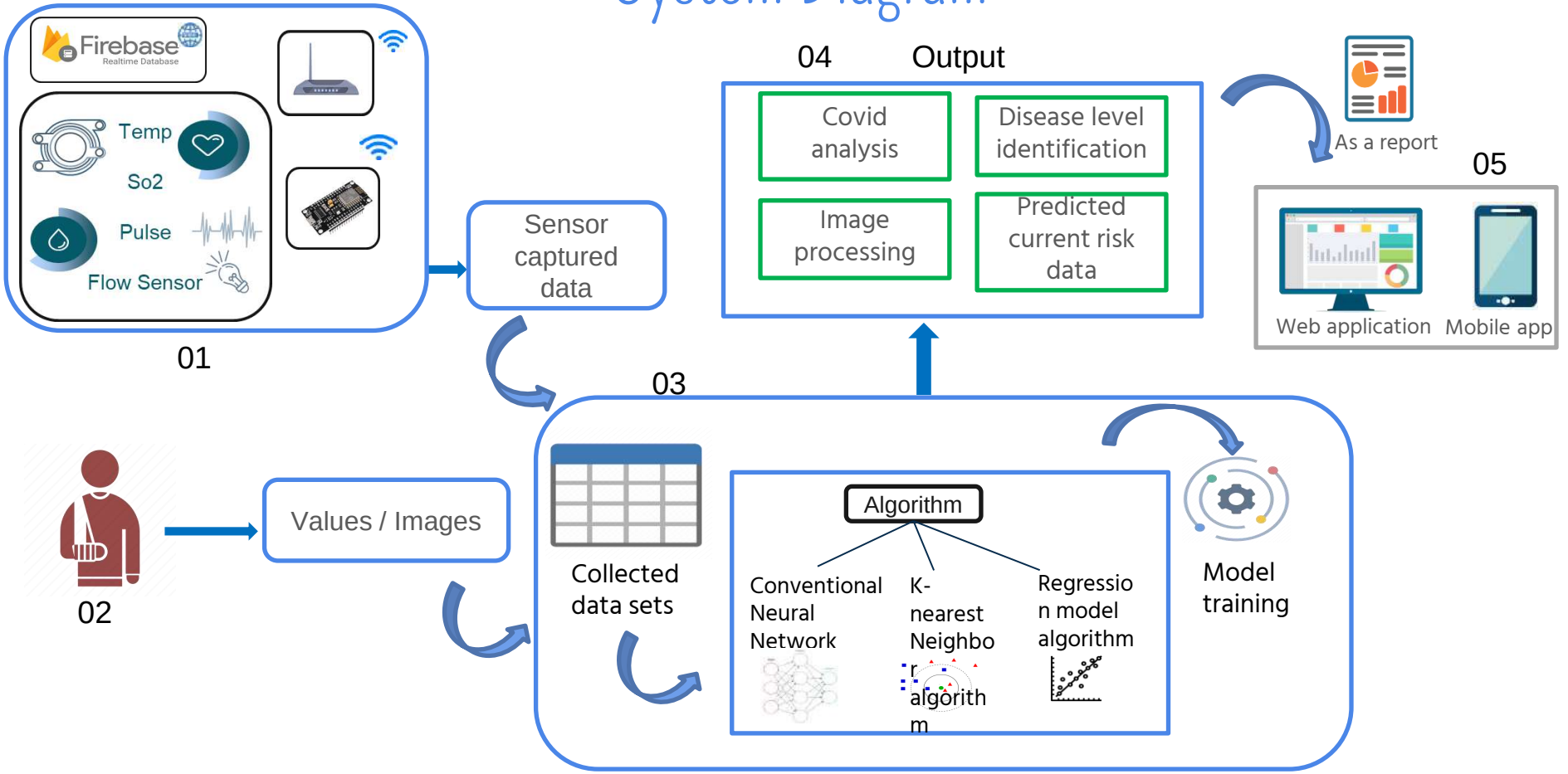
Overall solutions – 50%

- Developed an Arduino-based device that detects certain types of symptoms to diagnose certain heart and lung related diseases.
- Use some machine learning based techniques to identify diseases. and clarify it.
- Show diseases spread rate to the user.
- Developing a mobile application and web application to facilitate patient usage.

Research Objectives

- Implement a device to get parameters of the patient and identify Covid'19. (Possibility as a percentage)
- Disease level wise identification and provide suggestions/ recommendations to reduce the risk level.
- Identify the exact lung disease among other lung diseases.
- Identify the three major diseases spread rate in Sri Lanka.

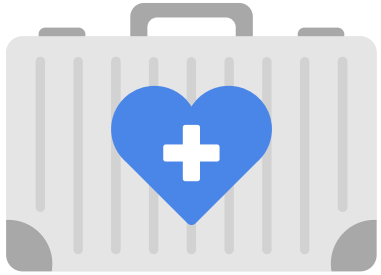
System Diagram



Focusing areas

Hospital /Medical centers

Schools



Offices

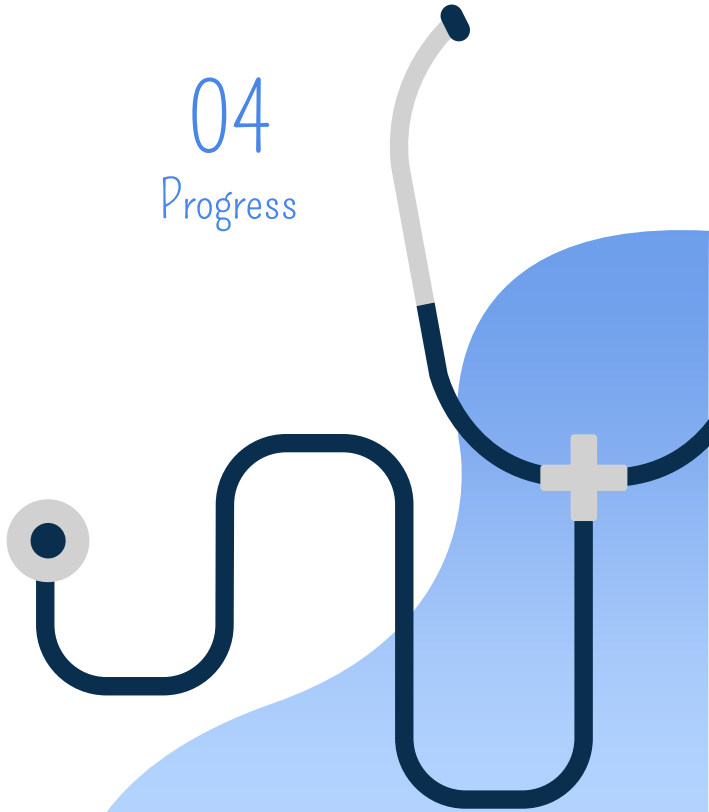
Crowded areas

IT18077698

Thennakoon T.M.B.C.K

Specialization | Information Technology

Table of Contents

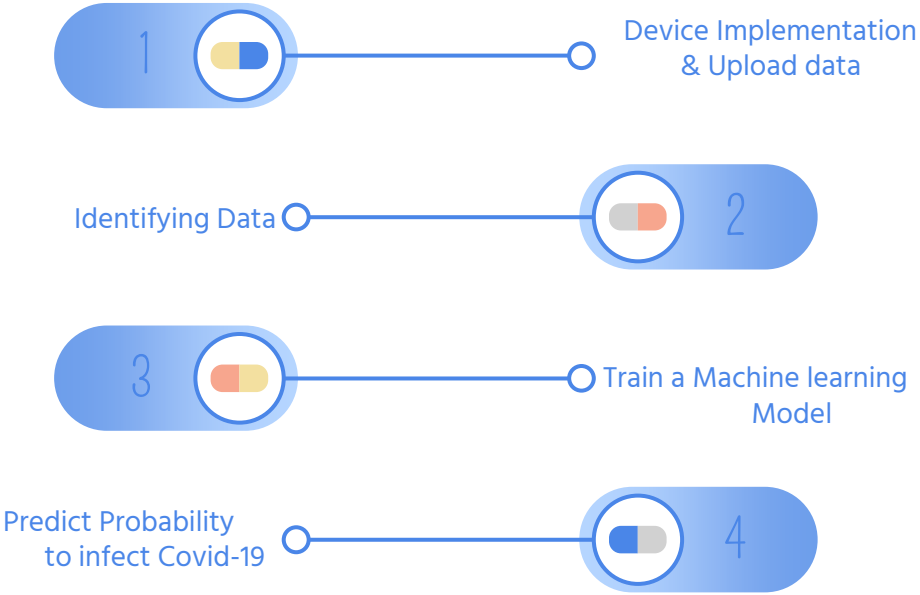


Research question

- Identify
  all measurements using single device with few minutes.
- Simple and user-friendly
  web application and mobile application.

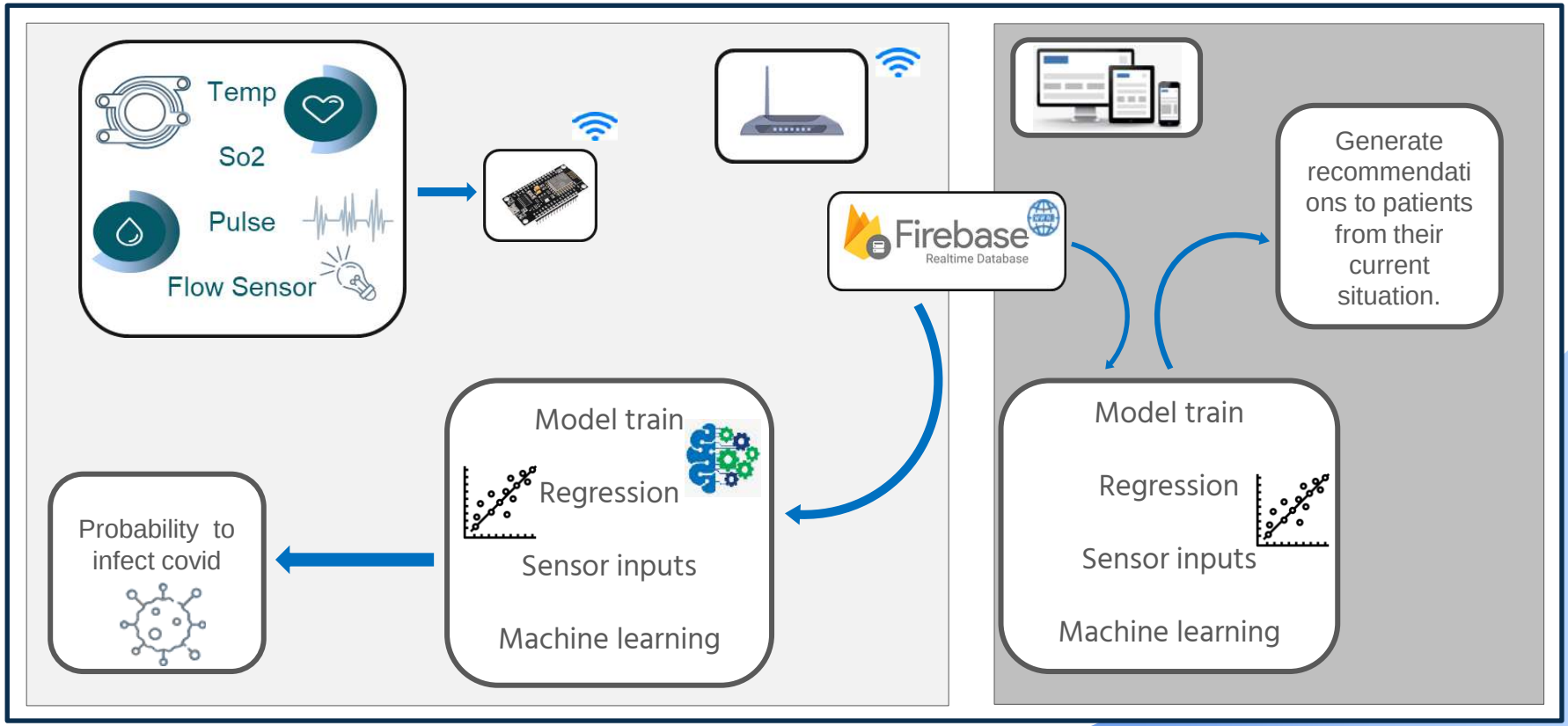
- Provide probability to infect Covid 19 & Give what are the necessary actions need to get by patient.
- Get necessary inputs and Generate healthy recommendatio ns to day-to-day life.

Achieved - 50%



IT18077698 | Thennakoon T M B C K| 2022-143

System diagram



Progress

- Implement device (Medisafe).
- Upload data to firebase.
- Developing a model.
- Give probability of infect Covid 19.

Completed(50%)

90% Progress presentation

- Completed Covisafe device.
- Generate recommendations to patients from their current situation.
- Completion of mobile application and web application.

- Completed device.
- Completed web & mobile application.
- Test all functions with patients and compare data.

Final Presentation

IT18077698 | Thennakoon T M B C K| 2022-143

Latest technologies in MediSafe

Medisafe device

ESP8266
Max30105sens
Flow meter
Firebase

Dataset collection and data training

Kaggle
Regression
Python

IDE

Jupiter Notebook
Arduino IDE
Visual studio code
Anaconda

Requirements

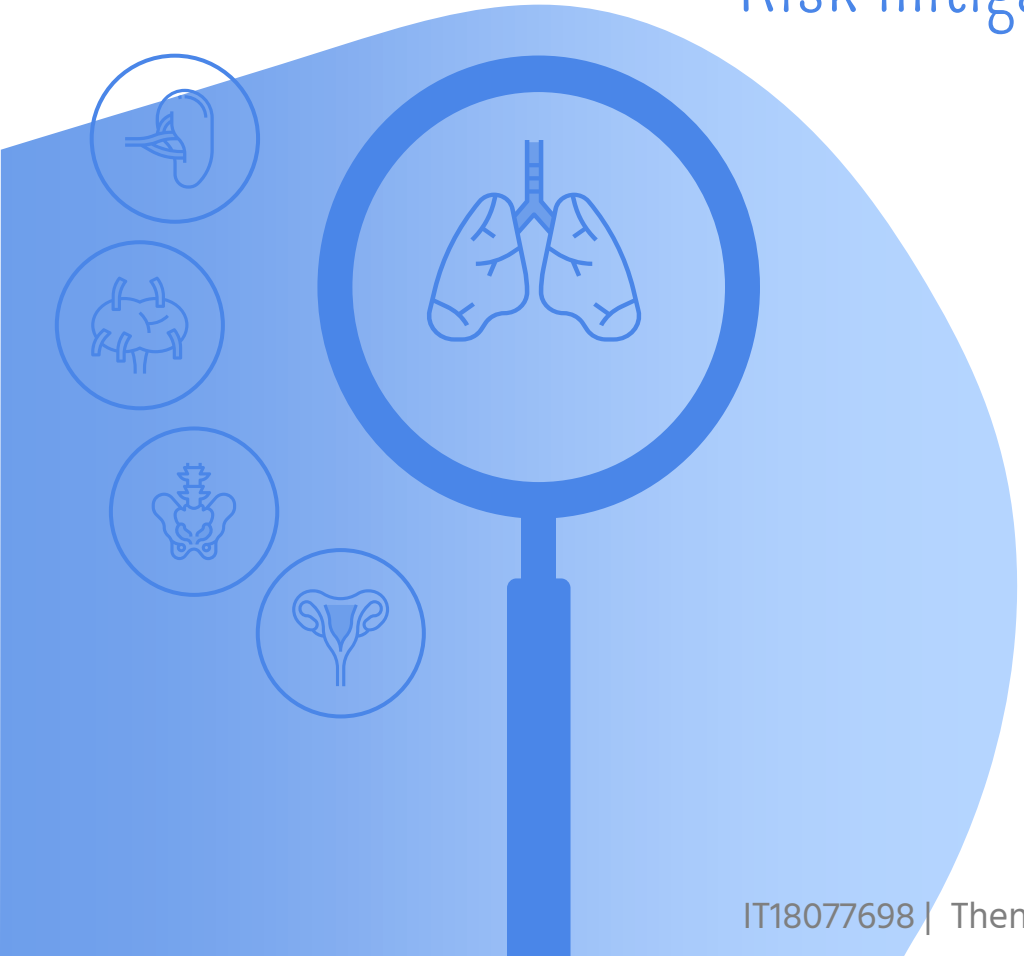
Functional

- Interoperability
- Accuracy
- Compliance

Non – functional

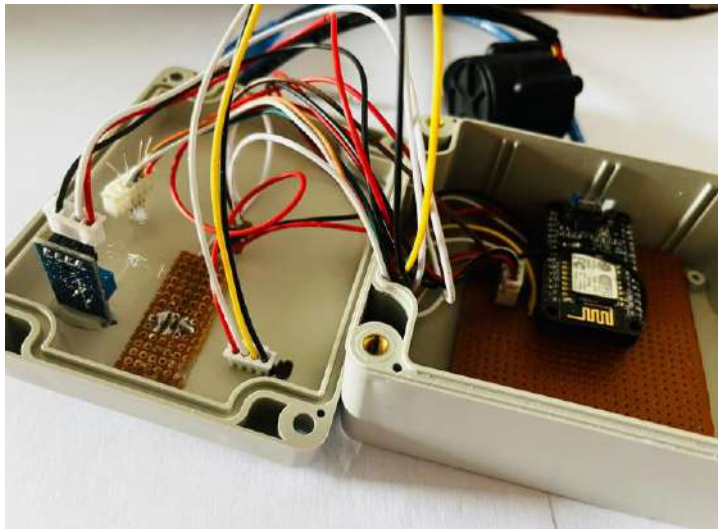
- Maintainability
- Manageability
- Usability
- Integrity

Risk mitigation



- Checking the manual again with the doctor for data obtained from the tool and what is provided in the processed data output.
- Periodically check the accuracy of the data obtained by the device
- Identifying hazards and hazardous situations associated with a medical device.

Completion of the project



Medisafe device

final2 | Arduino 1.8.19 (Windows Store 1.8.57.0)

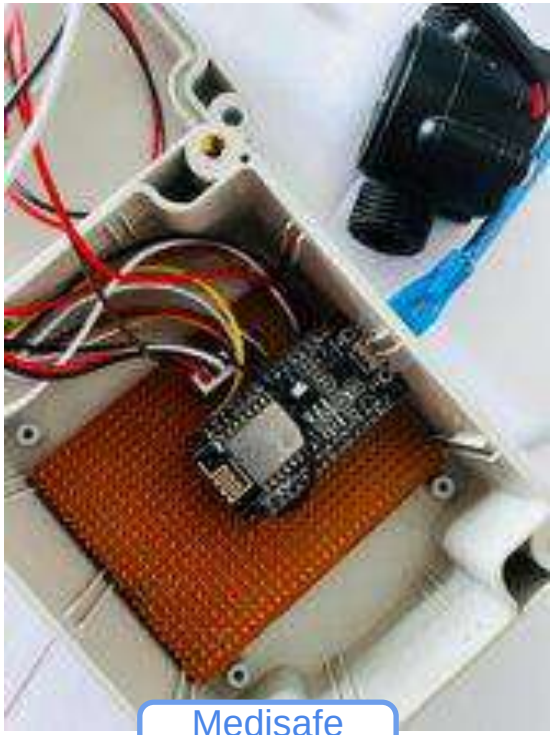
File Edit Sketch Tools Help

final2 § | Firebase | flow_meater | max30105 | so2 | temp

```
int count = 0;
int temp_bps;
//---------------------Firebase-----------

#include <ArduinoJson.h>
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>
// Set these to run example.
#define FIREBASE_HOST "medisafe-research-default-rtdb.firebaseio.com/unit_1"
#define FIREBASE_AUTH "qjnABtFp7TrCzENApcxBGQSeI21kghAc1OPwrBB5"
#define WIFI_SSID "supun"
#define WIFI_PASSWORD "supun111191"
FirebaseData firebaseData;

#define SENSOR  D4
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
//boolean ledState = LOW;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
// ------------------led

#define REDLED  D5
```

Arduino IDE

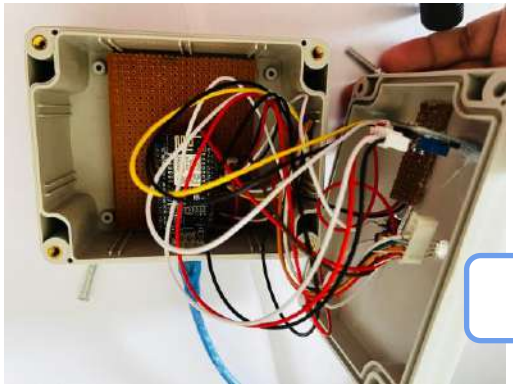
Completion of the project



Medisafe device

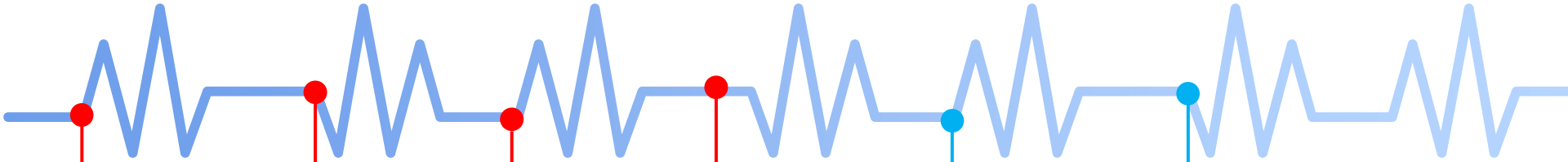
```
C:\WINDOWS\system32\cmd.exe

(covid) C:\Users\user>cd C:\Users\user\Desktop\24-04-2022\covid

(covid) C:\Users\user\Desktop\24-04-2022\covid>C:

(covid) C:\Users\user\Desktop\24-04-2022\covid>python Runcovid.py
type oxygen level : 90
type your pulse : 96
type your Temperature : 90
confidence : 100.0 %
The probability of having a covid infection is 35.360000000000004%
Traceback (most recent call last):
  File "Runcovid.py", line 1, in <module>
    from covid import predictc
ImportError: cannot import name 'predictc' from 'covid' (C:\Users\user\Desktop\24-04-2022\covid\covid.py)

(covid) C:\Users\user\Desktop\24-04-2022\covid>
```

Covid prediction output



Medisafe device

Self work breakdown structure



Phase 1

Gather relevant sensors & microprocessor

Phase 2

Intergrade device and implement

Phase 3

Find datasets and train model

Phase 4

Update device data in database and give covid probability

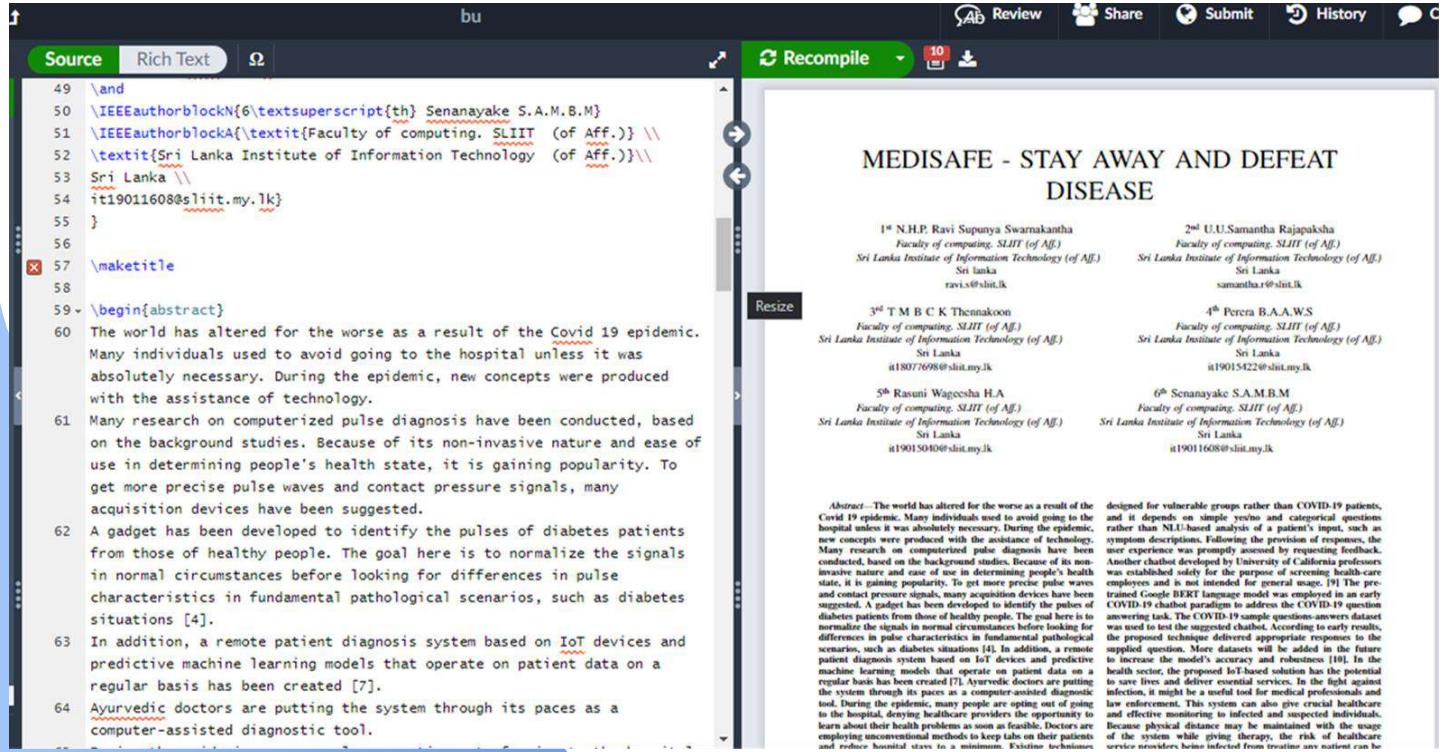
Phase 5

Generate recommendations For day-to-day life and specific areas patient need to attend.

Phase 6

Intergrade device and system , web application , mobile application.

Individual overleaf conference paper

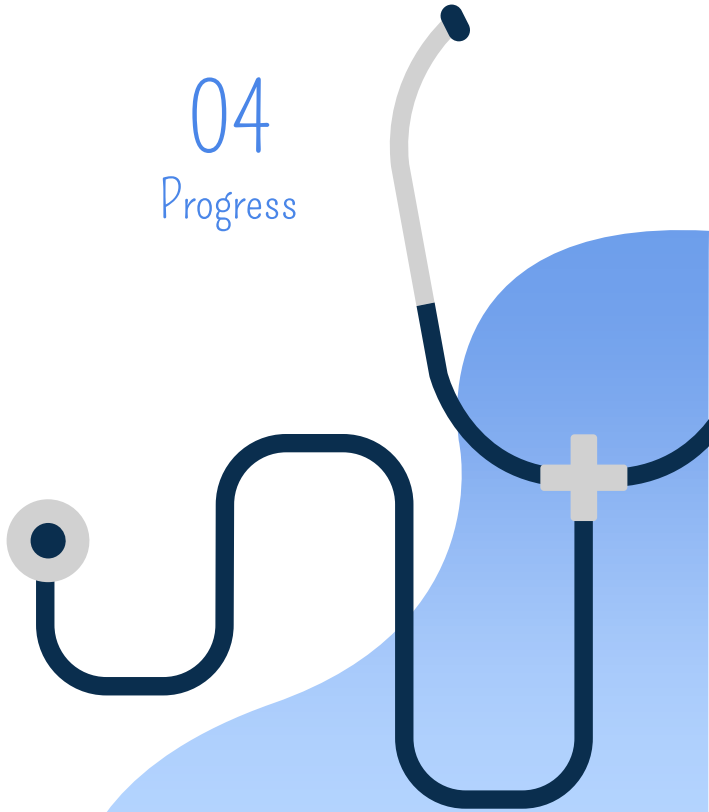


IT19015422

Perera B.A.A.W.S

Specialization | Information Technology

Table of Contents

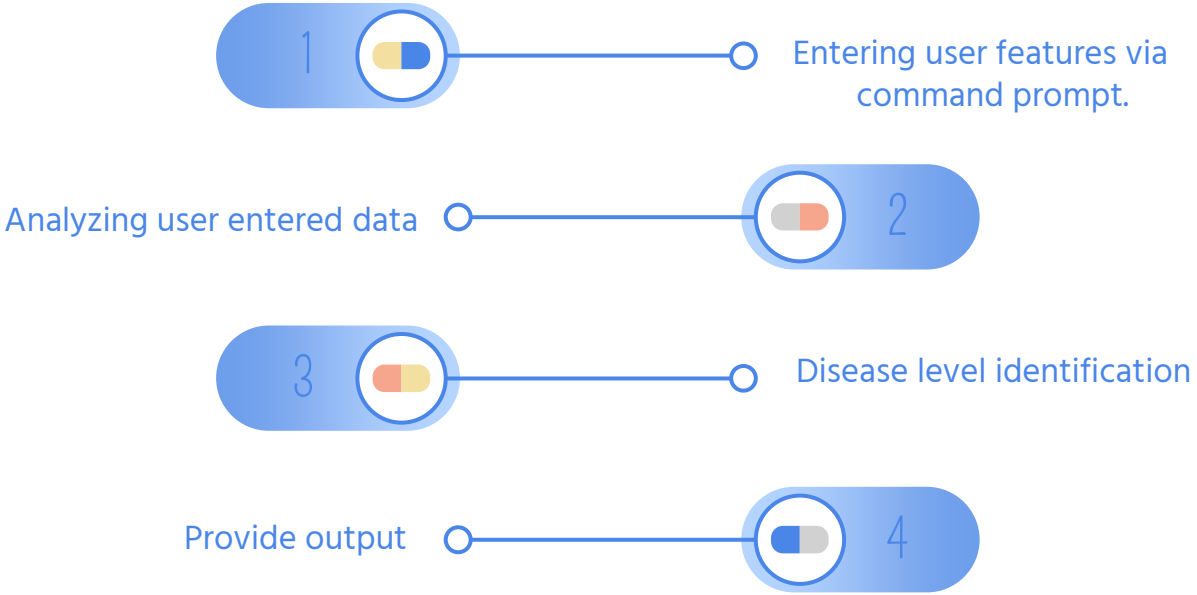


Research question

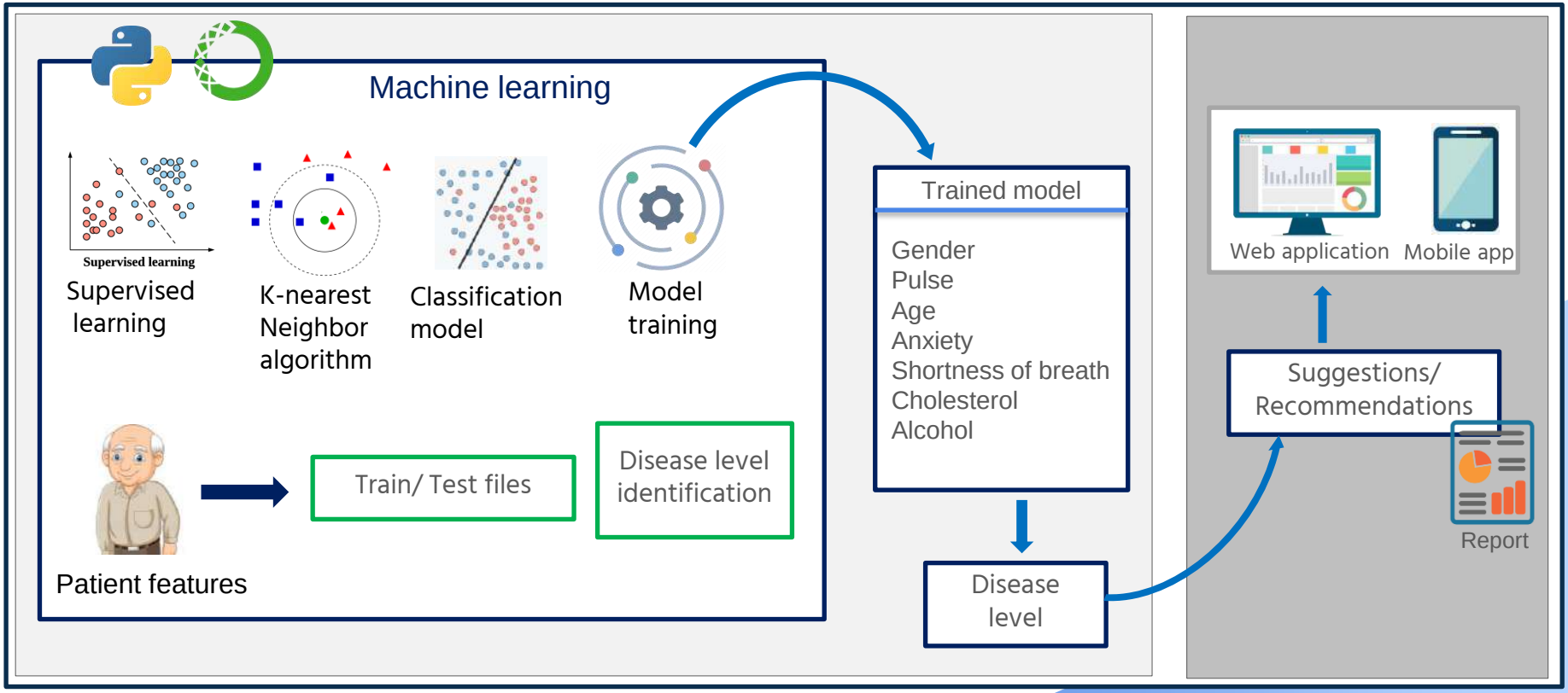
- How to identify the people who are suffering in such lung and heart diseases (level wise)
- How to check current situation in cost effectively

- How to provide easily recommendations via web/ mobile application to the user
- What are the solutions we can give due to shortage of medicines

Achieved – 50%


1
Entering user features via command prompt.
Analyzing user entered data
2
3
Disease level identification
Provide output
4

System diagram



Progress

- Study the technology
- Data collecting
- Data analysis
- Find proper algorithm (K-nearest neighbors)
- Train the model
- Get the output as level wise related to the disease

Completed(50%)

90% Progress presentation

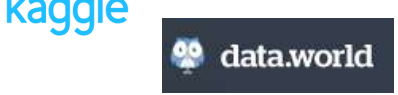
- Provide suggestions and recommendations to the user
- Web application implementation
- Mobile application implementation

- Completion of mobile application
- Completion of web application
- Integrate member components together

Final Presentation

Latest technologies in MediSafe

DataSet collection



- https://data.world/informatics-edu/heart-disease-prediction
- https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Model training

- K – nearest neighbor algorithm
- Libraries – pandas, sklearn, joblib, numpy …..

IDE

- Vs Code
- Jupyter notebook
- Anaconda prompt

Requirements

Functional

- Interoperability
- Authentication.
- Report generate
- User friendly

Non - functional

- Quality
- Durability
- Security
- Privacy

Risk mitigation

- ✓ Entering current situation features difficult to known by person . So that those features will get from the implemented device. (In future – 90%)
- ✓ Adults are not well fluent in new technologies.
- ✓ Validity of the disease level will depend on the user inputs.

Completion of the project

Model training

predict.py

```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
import joblib as jb
import pandas as pd

dataframe_H = pd.read_csv('Heart_attack.csv')
dataframe_P = pd.read_csv('Pneumonia.csv')
dataframe_W = pd.read_csv('Wheezing.csv')

namesH = [
    'age', 'Gender', 'Cholesterol', 'Pulse', 'Smoke', 'Alcohol', 'Risk'
]

namesP = [
    'age', 'Gender', 'Shortness_of_breath', 'Pulse', 'Smoke', 'Alcohol', 'Risk'
]

namesW = [
    'age', 'Gender', 'Anxiety', 'Shortness_of_breath', 'Smoke', 'Alcohol', 'Risk'
]

XH = dataframe_H.iloc[:, :-1].values
yH = dataframe_H.iloc[:, 6].values

XP = dataframe_P.iloc[:, :-1].values
yP = dataframe_P.iloc[:, 6].values

XW = dataframe_W.iloc[:, :-1].values
yW = dataframe_W.iloc[:, 6].values

X_trainH, X_testH, y_trainH, y_testH = train_test_split(XH, yH, test_size=0.20)
X_trainP, X_testP, y_trainP, y_testP = train_test_split(XP, yP, test_size=0.20)
```

run.py 4 | predict.py 5

predict.py

```python
X_trainH, X_testH, y_trainH, y_testH = train_test_split(XH, yH, test_size=0.20)
X_trainP, X_testP, y_trainP, y_testP = train_test_split(XP, yP, test_size=0.20)
X_trainW, X_testW, y_trainW, y_testW = train_test_split(XW, yW, test_size=0.20)

scalerH = StandardScaler()
scalerP = StandardScaler()
scalerW = StandardScaler()

scalerH.fit(X_trainH)
scalerP.fit(X_trainP)
scalerW.fit(X_trainW)

X_trainH = scalerH.transform(X_trainH)
X_testH = scalerH.transform(X_testH)

X_trainP = scalerP.transform(X_trainP)
X_testP = scalerP.transform(X_testP)

X_trainW = scalerW.transform(X_trainW)
X_testW = scalerW.transform(X_testW)


classifierH = KNeighborsClassifier(n_neighbors=6)
classifierP = KNeighborsClassifier(n_neighbors=6)
classifierW = KNeighborsClassifier(n_neighbors=6)


classifierH.fit(X_trainH, y_trainH)
classifierP.fit(X_trainP, y_trainP)
classifierW.fit(X_trainW, y_trainW)


scaler_fileH = "scalerH.save"
```

run.py 4 | predict.py 5

predict.py

```python
classifierH.fit(X_trainH, y_trainH)
classifierP.fit(X_trainP, y_trainP)
classifierW.fit(X_trainW, y_trainW)


scaler_fileH = "scalerH.save"
scaler_fileP = "scalerP.save"
scaler_fileW = "scalerW.save"

model_fileH = "model_fileH.save"
model_fileP = "model_fileP.save"
model_fileW = "model_fileW.save"

jb.dump(scalerH, scaler_fileH)
jb.dump(scalerP, scaler_fileP)
jb.dump(scalerW, scaler_fileW)

jb.dump(classifierH, model_fileH)
jb.dump(classifierP, model_fileP)
jb.dump(classifierW, model_fileW)


# y_predict = classifier.predict(X_test)

# # Print results:
# print(confusion_matrix(y_test, y_predict))
# print(classification_report(y_test, y_predict))
```

Completion of the project

Level implementation

predict.py 5

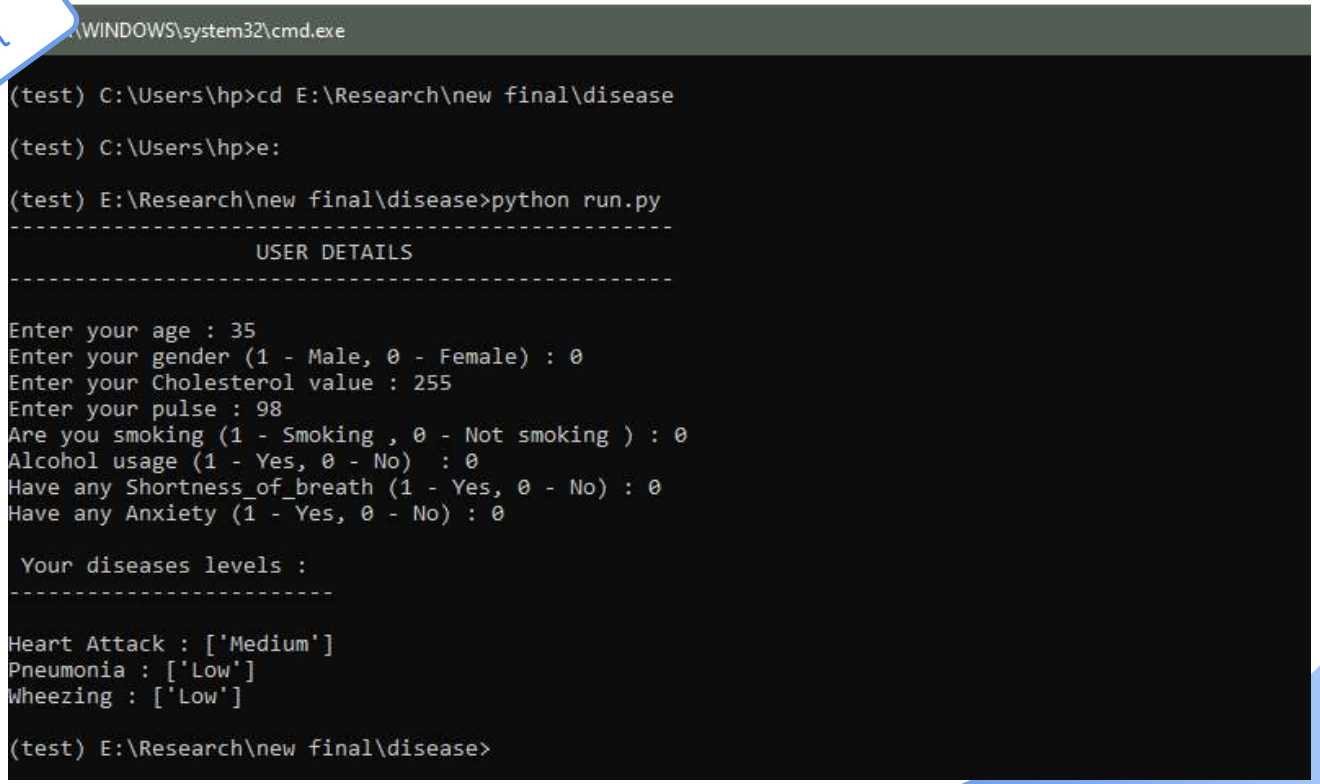
```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
#from sklearn.metrics import classification_report, confusion_matrix
import joblib as jb
import pandas as pd

scaler_fileH = "scalerH.save"
scaler_fileP = "scalerP.save"
scaler_fileW = "scalerW.save"

model_fileH = "model_fileH.save"
model_fileP = "model_fileP.save"
model_fileW = "model_fileW.save"

scalerH = jb.load(scaler_fileH)
scalerP = jb.load(scaler_fileP)
scalerW = jb.load(scaler_fileW)

classifierH = jb.load(model_fileH)
classifierP = jb.load(model_fileP)
classifierW = jb.load(model_fileW)

```

run.py 4 | predict.py 5

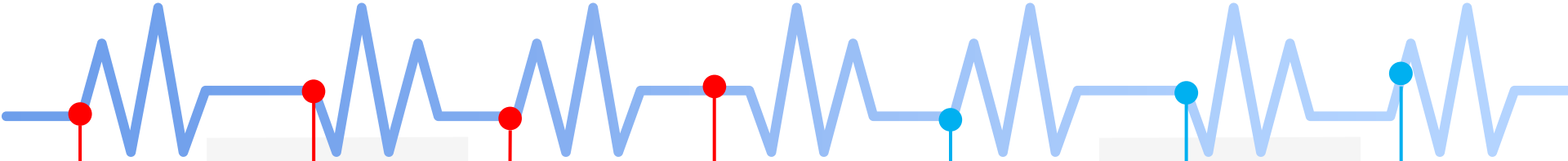
```python
print('-------------------------------------------------')
print('                  USER DETAILS                   ')
print('-------------------------------------------------')
print('                                                 ')
age = input('Enter your age : ')
Gender = input('Enter your gender (1 - Male, 0 - Female) : ')
Cholesterol = input('Enter your Cholesterol value : ')
Pulse = input('Enter your pulse : ')
Smoke = input('Are you smoking (1 - Smoking , 0 - Not smoking ) : ')
Alcohol = input('Alcohol usage (1 - Yes, 0 - No)  : ')
Shortness_of_breath = input('Have any Shortness_of_breath (1 - Yes, 0 - No) : ')
Anxiety = input('Have any Anxiety (1 - Yes, 0 - No) : ')

valH = [age, Gender, Cholesterol, Pulse, Smoke, Alcohol]
valP = [age, Gender, Shortness_of_breath, Pulse, Smoke, Alcohol]
valW = [age, Gender, Anxiety, Shortness_of_breath, Smoke, Alcohol]
valH = scalerH.transform([valH])
valP = scalerH.transform([valP])
valW = scalerH.transform([valW])
y_predictH = classifierH.predict(valH)
y_predictP = classifierH.predict(valP)
y_predictW = classifierH.predict(valW)
print('')
print(' Your diseases levels :')
print('-------------------------')
print('')
print('Heart Attack :',y_predictH)
print('Pneumonia :',y_predictP)
print('Wheezing :',y_predictW)

```

Completion of the project

Output



Self work breakdown structure



Phase 1

Learning about related technologies

Phase 2

Collect data sets

Phase 3

Model training

Phase 4

Disease level wise identification

Phase 5

Provide suggestions and recommendations (Report generate)

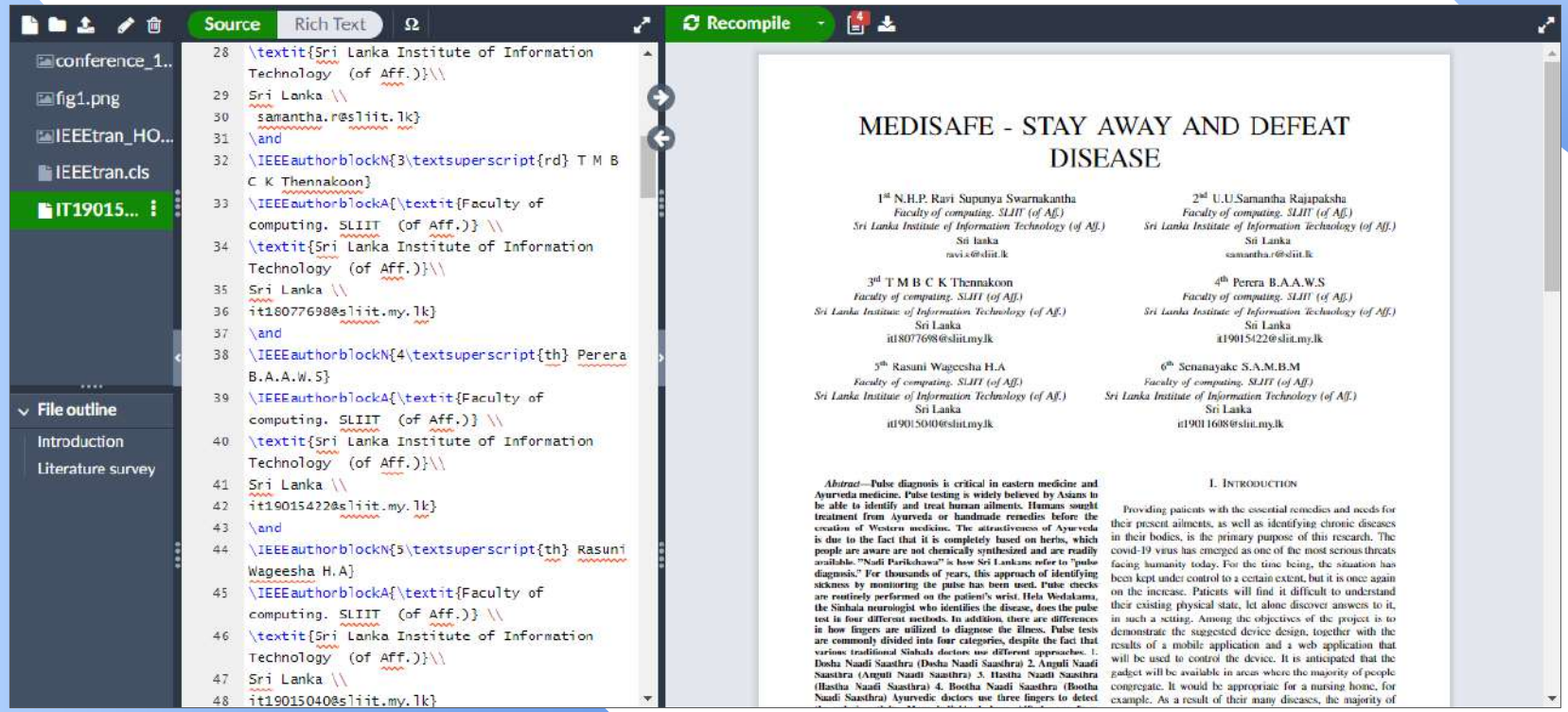
Phase 6

Web application / Mobile app implementation

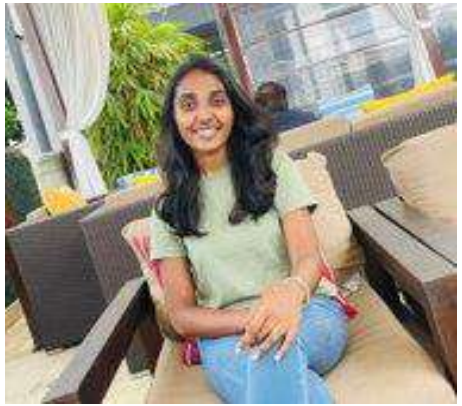
Phase 7

Integration

Individual overleaf conference paper



https://www.overleaf.com/project/626e65c27e9110853ed631c7

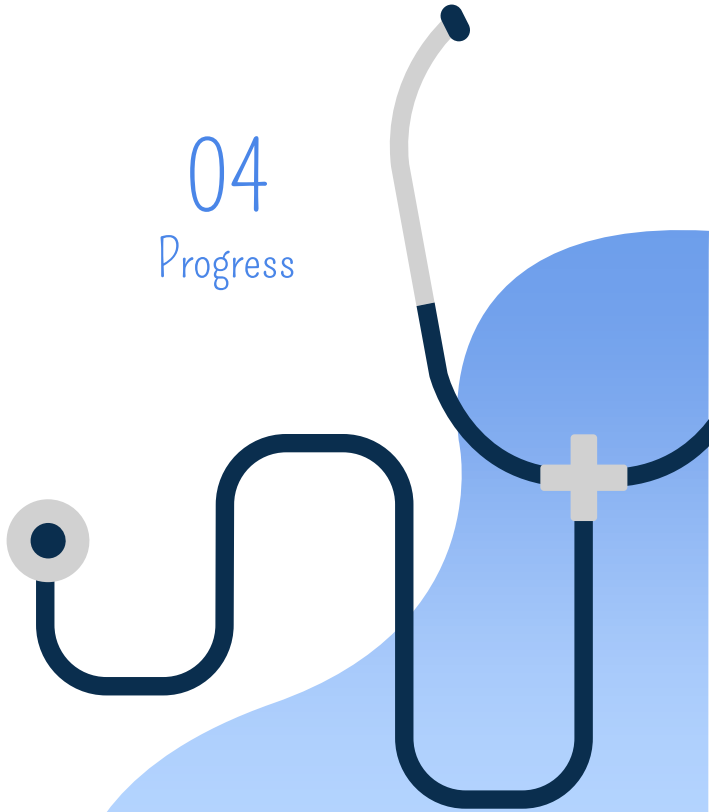


IT19015040

Rasuni Wageesha H.A

Specialization | Information Technology

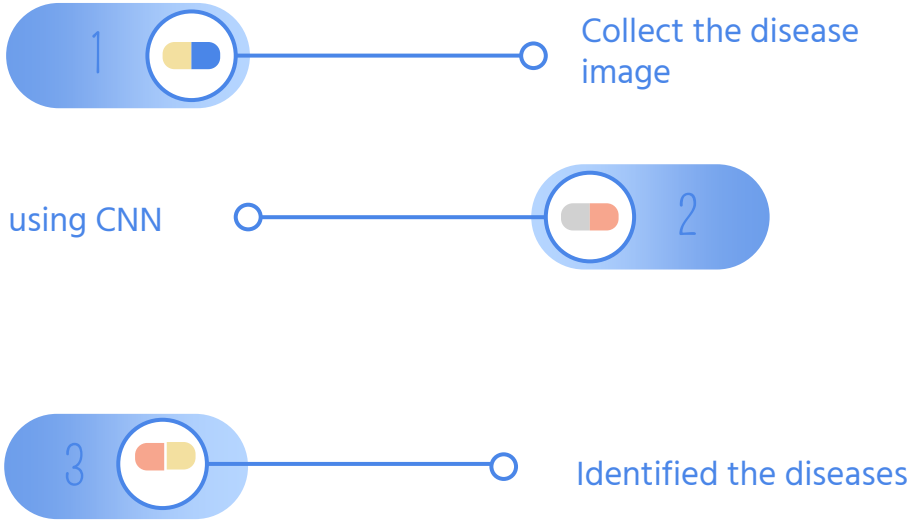
Table of Contents



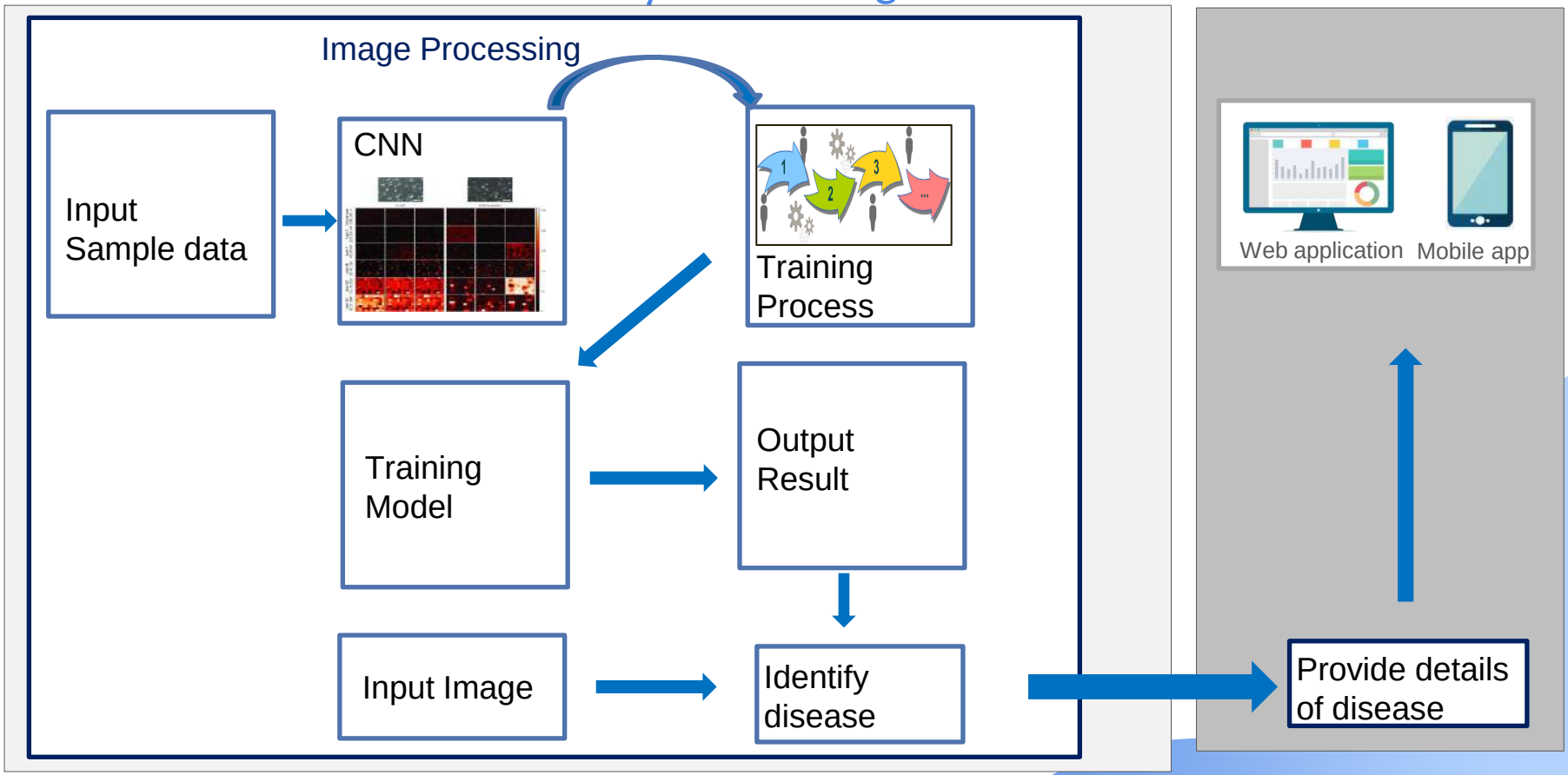
Research question

There are many different types of lung diseases and diagnosing one might be difficult.

Achieved – 50%


1
Collect the disease image
Train image using CNN
2
3
Identified the diseases

System diagram



Progress

- Collecting diseases images.
- Identify how to develop the system.
- Trained images and generated a model using CNN.
- Got the output using trained module for test data.

Completed(50%)

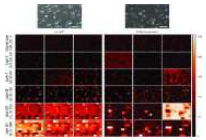
90% Progress presentation

- Provide details of the disease.
- Web application implementation.
- Mobile application implementation.

- Completion of mobile application.
- Completion of web application.
- Integrate member co

Final Presentation

Technologies in MediSafe

Collected data	Algorithem	IDE
kaggle	 TensorFlow	
https://www.kaggle.com/datasets/mdsafayet/multiclass-disease-classification	Convolutional Neural Network Tesor flow	Visual Studio

Requirements

Functional

- Upload the lung image to the system.

Non – functional

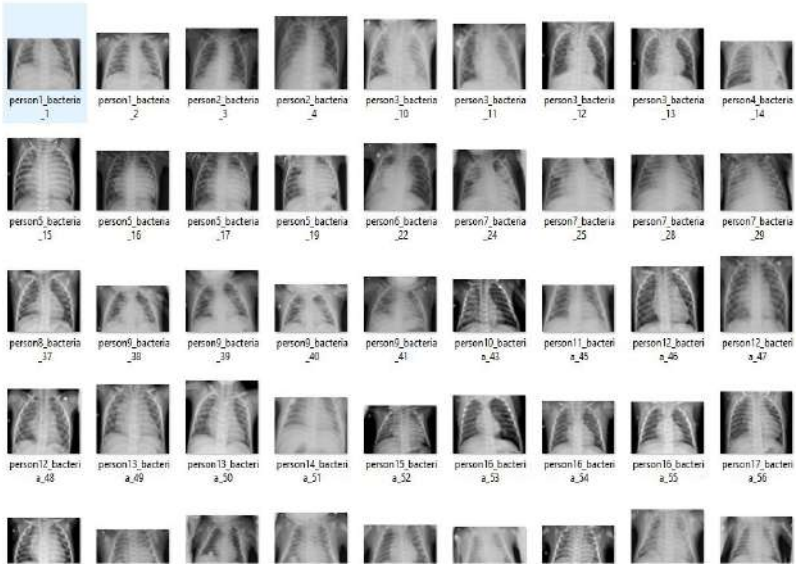
- Performance
- Availability
- Reliability

Risk mitigation

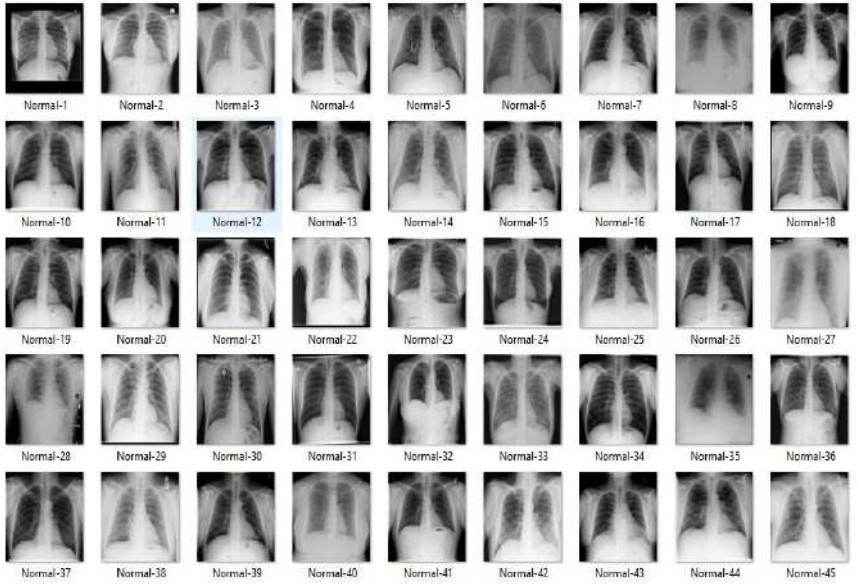
- ✓ Validity of the clear images.

Completion of the project

Normal chest X rays

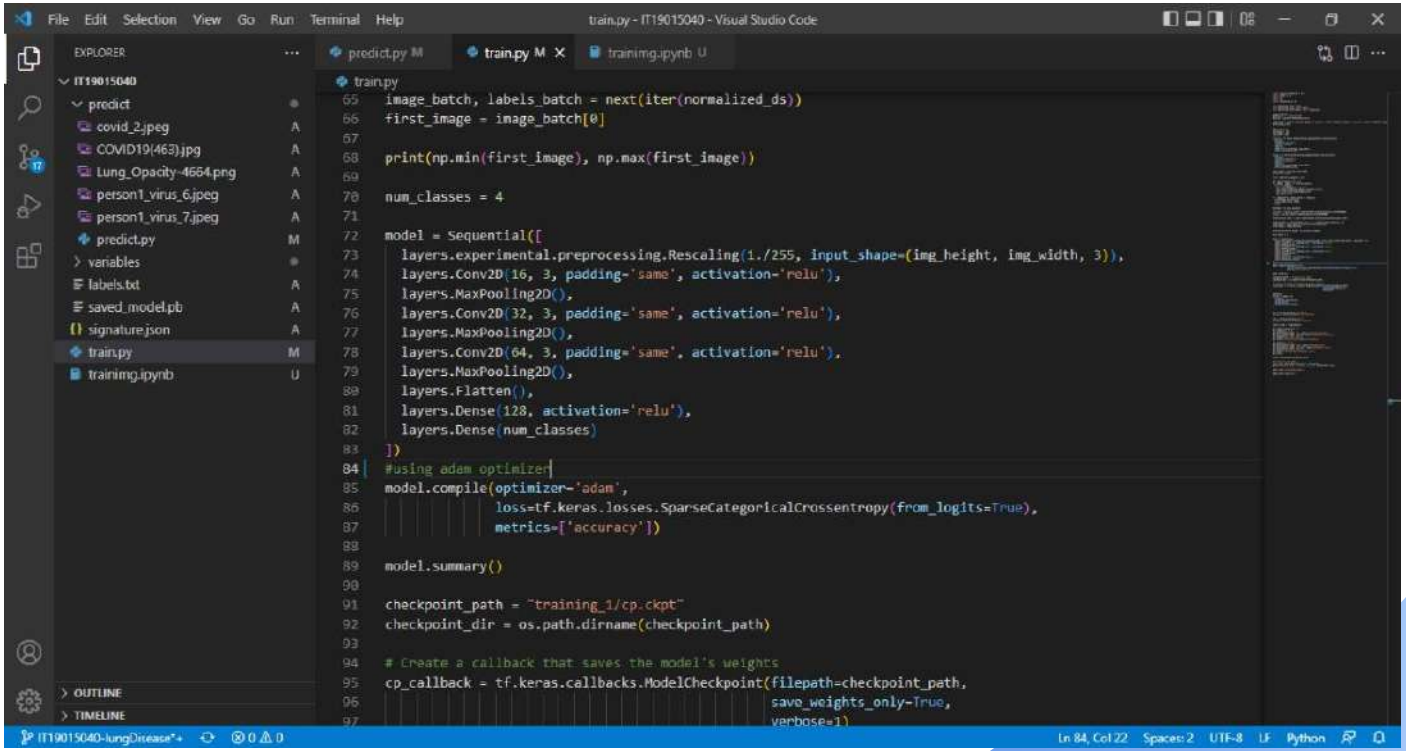


PNEUMONIA Chest Xray



Completion of the project

Model training



Completion of the project

Model training

trainimg.ipynb

File Edit View Insert Runtime Tools Help Last saved at 21:19

Comment Share

Table of contents

Section

+ Code + Text

Connect Editing

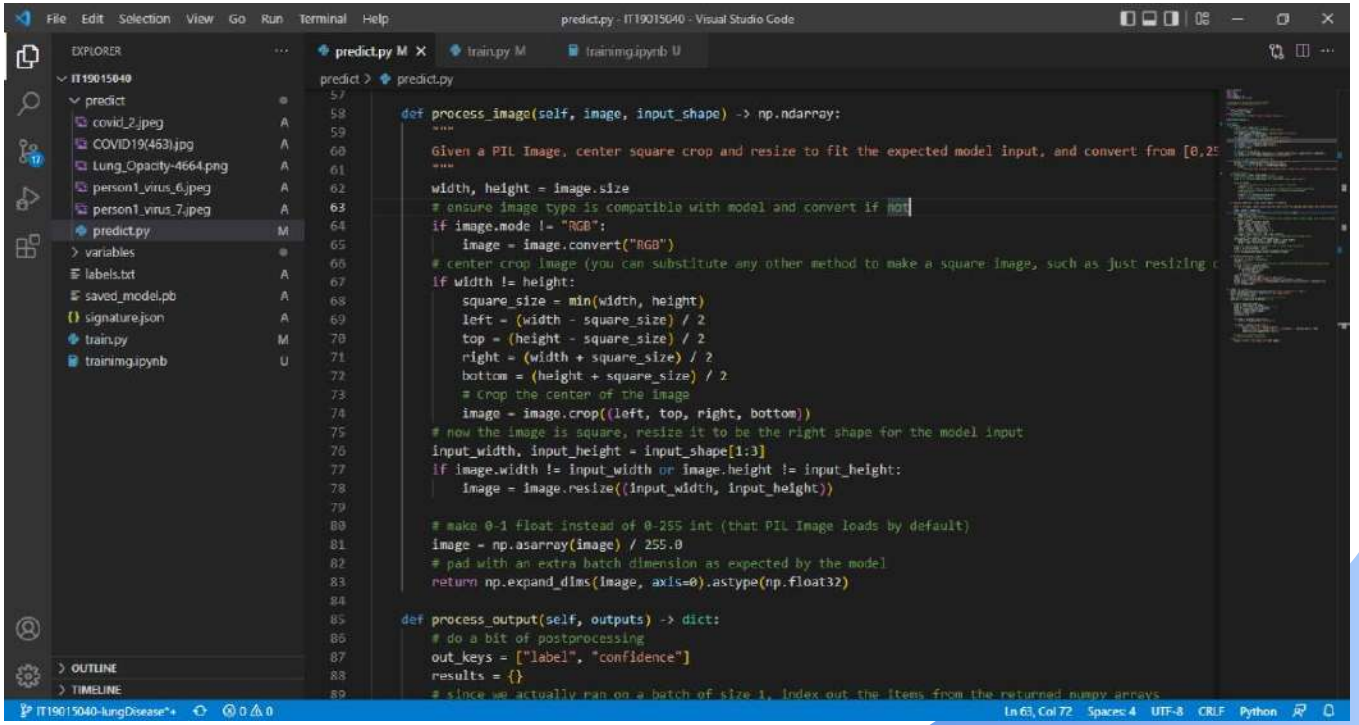
```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] covid = list(data_dir.glob('covid/*'))
    PIL.Image.open(str(covid[0]))
```

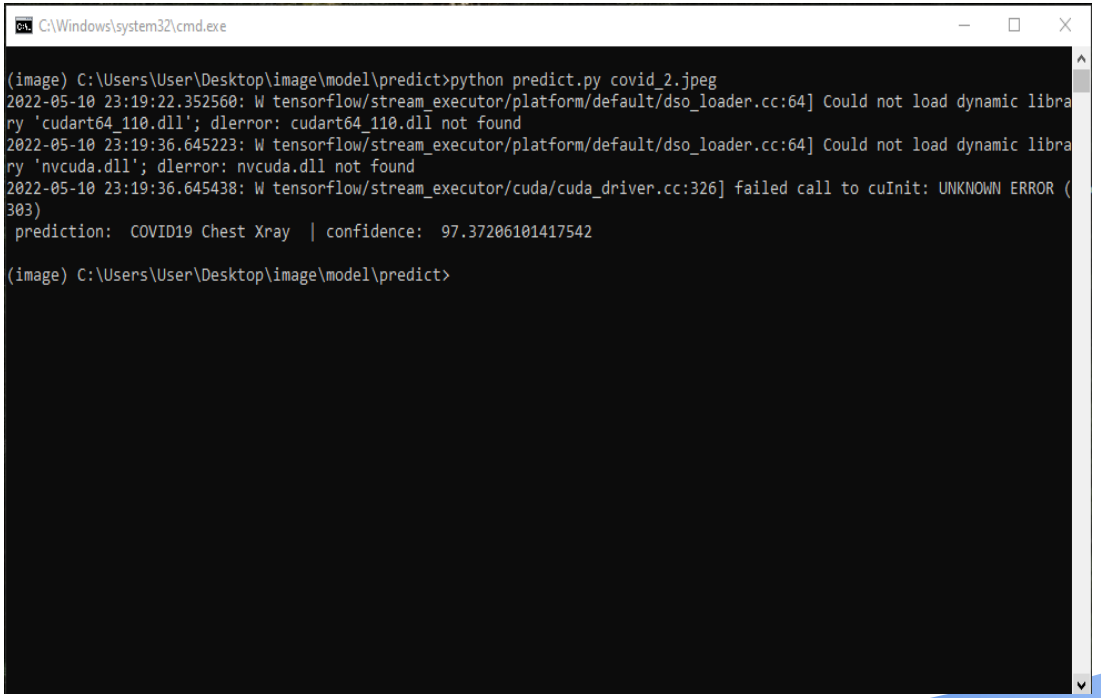
Completion of the project

Pre-processing

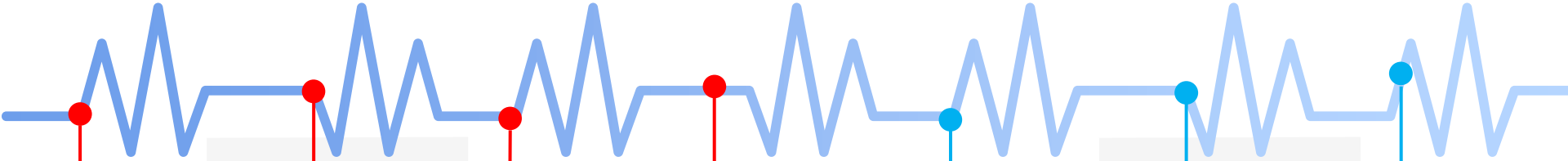


Completion of the project

Output



Self work breakdown structure



Phase 1

Learning about related technologies

Phase 2

Collect data sets

Phase 3

Model training

Phase 4

Identify the diseases

Phase 5

Provide details of the diseases.

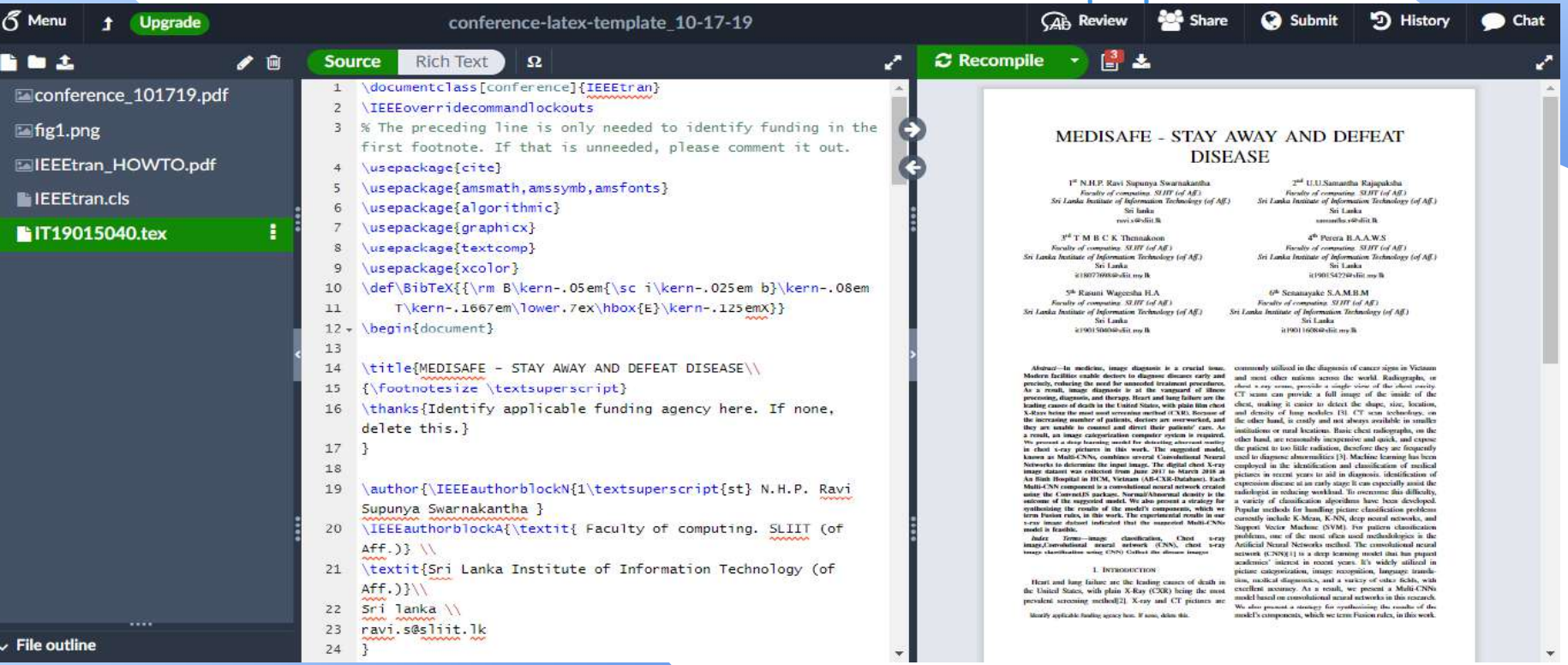
Phase 6

Web application / Mobile app implementation

Phase 7

Integration

Individual overleaf conference paper



https://www.overleaf.com/project/626ea0e7a31e802185bd3e61

IT19011608

Senanayaka S.A.M.A.B.M

Specialization | Information Technology

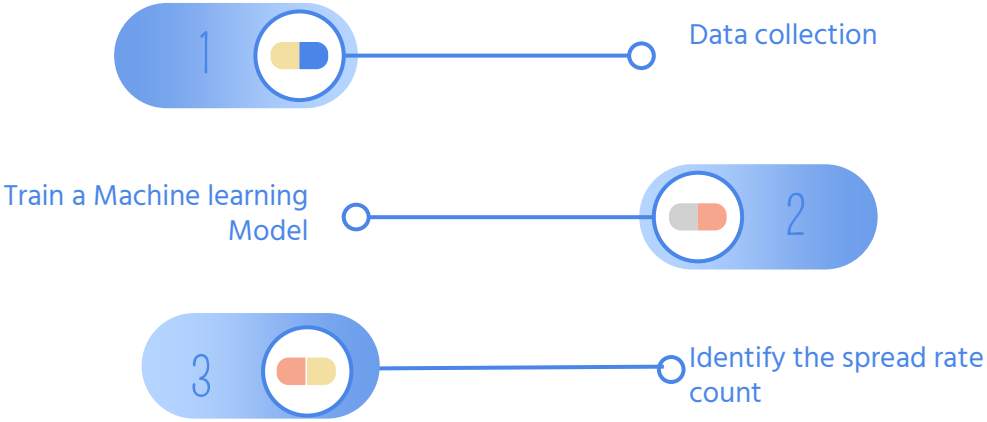
Table of Contents

Research question

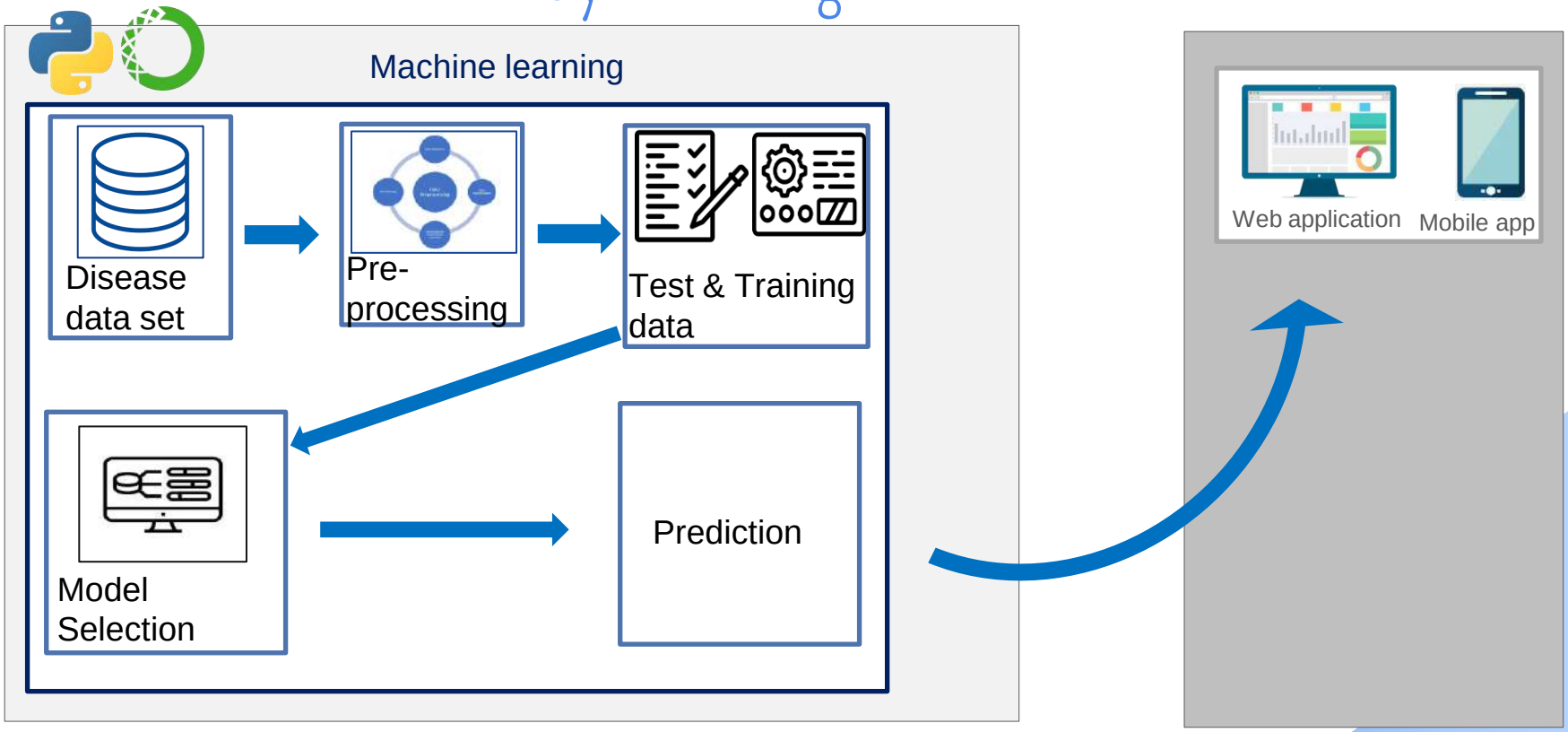
- Identify the diseases count on the sri lanka

- Target domain

Achieved – 50%


1
Data collection
Train a Machine learning Model
2
3
Identify the spread rate count

System diagram



Progress

- Study the technology
- Data collecting
- Data analysis
- Find proper algorithm
- Train the model
- identify the spread rate count

Completed(50%)

90% Progress presentation

- Create API connection
- Web application implementation
- Mobile application implementation

- Completion of mobile application
- Completion of web application
- Integrate member components together

Final Presentation

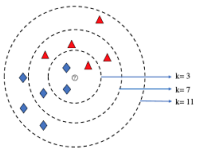
Technologies in MediSafe

Collected data

kaggle

1) https://www.healthdata.org/sri-lanka
2) https://www.kaggle.com/

Algorithem



Long short term-memory algorithms

Ide

jupyter

Jupiter notebook and vs code

Requirements

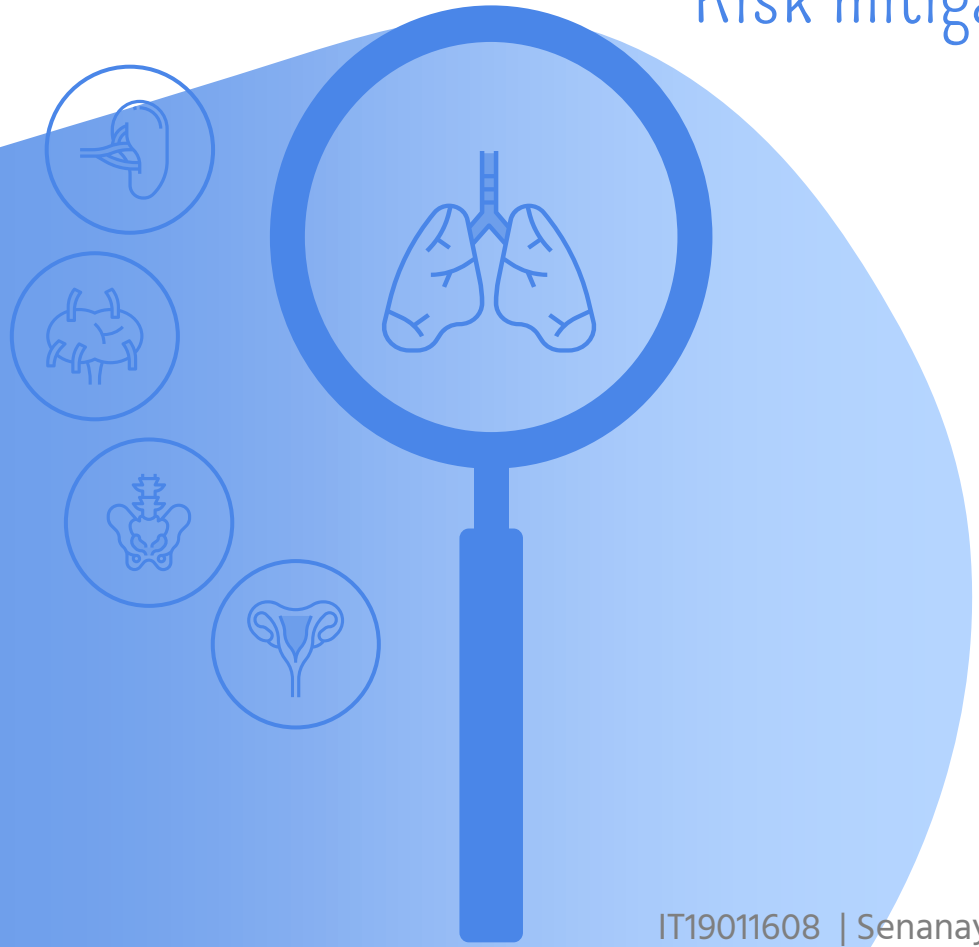
Functional

- Identify the spread rate count
- Display the data healthcare dashboard

Non – functional

- Accuracy
- Availability

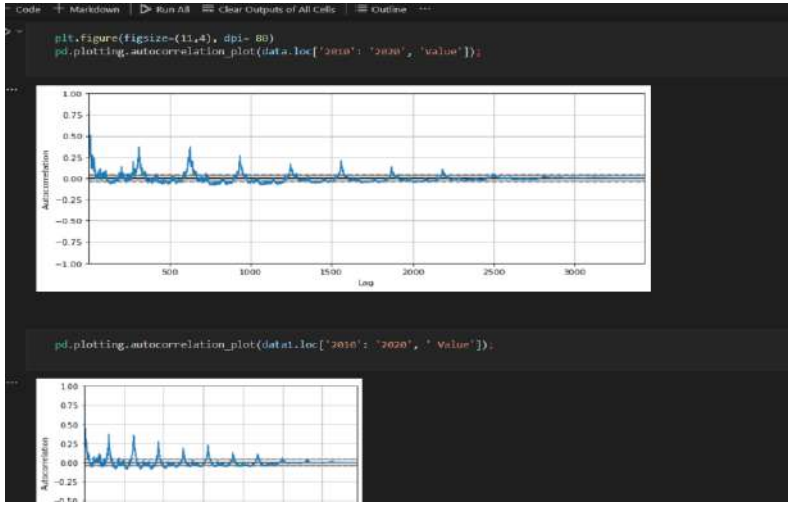
Risk mitigation



Find relevant and accurate data

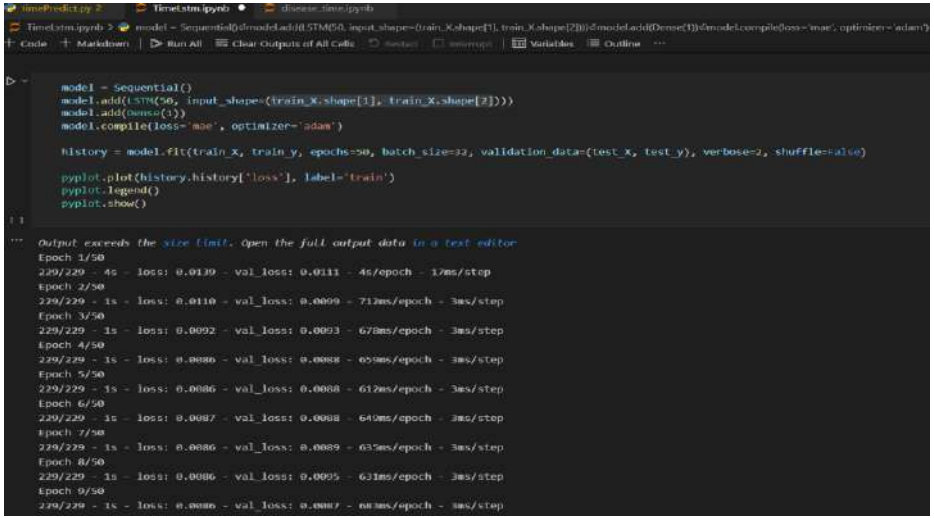
Completion of the project

Pre- processing stage



Completion of the project

Model training stage

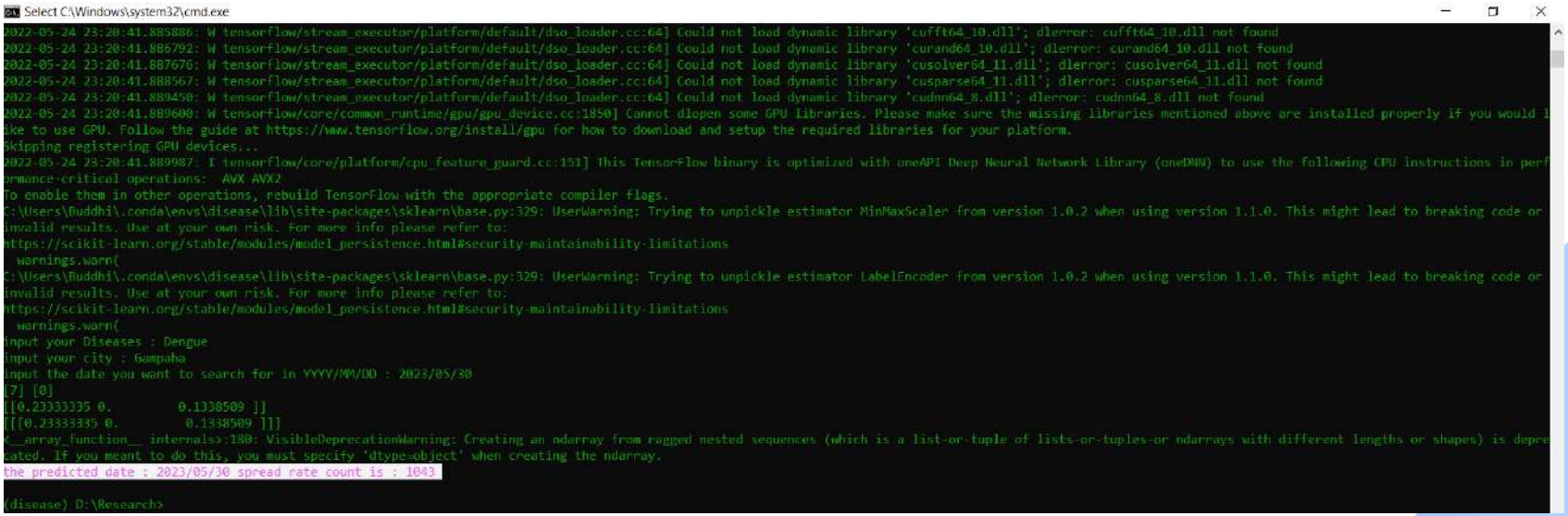


```
model = Sequential()
model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')

history = model.fit(train_X, train_y, epochs=50, batch_size=32, validation_data=(test_X, test_y), verbose=2, shuffle=False)

pyplot.plot(history.history['loss'], label='train')
pyplot.legend()
pyplot.show()
```

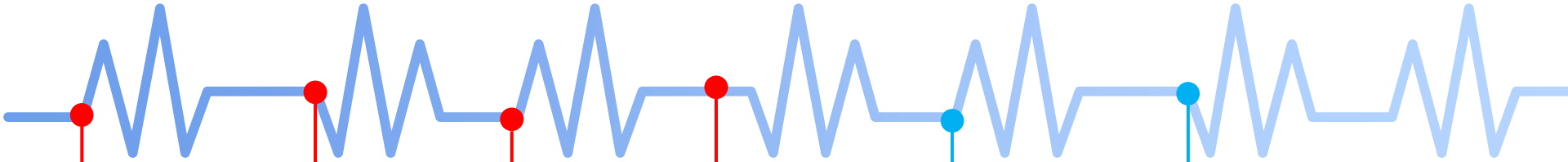
```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/50
229/229 - 4s - loss: 0.0139 - val_loss: 0.0111 - 4s/epoch - 17ms/step
Epoch 2/50
229/229 - 1s - loss: 0.0110 - val_loss: 0.0099 - 712ms/epoch - 3ms/step
Epoch 3/50
229/229 - 1s - loss: 0.0092 - val_loss: 0.0093 - 678ms/epoch - 3ms/step
Epoch 4/50
229/229 - 1s - loss: 0.0086 - val_loss: 0.0088 - 659ms/epoch - 3ms/step
Epoch 5/50
229/229 - 1s - loss: 0.0086 - val_loss: 0.0088 - 612ms/epoch - 3ms/step
Epoch 6/50
229/229 - 1s - loss: 0.0087 - val_loss: 0.0088 - 640ms/epoch - 3ms/step
Epoch 7/50
229/229 - 1s - loss: 0.0086 - val_loss: 0.0089 - 635ms/epoch - 3ms/step
Epoch 8/50
229/229 - 1s - loss: 0.0086 - val_loss: 0.0095 - 631ms/epoch - 3ms/step
Epoch 9/50
229/229 - 1s - loss: 0.0086 - val_loss: 0.0087 - 603ms/epoch - 3ms/step
```

Completion of the project

Output



Self work breakdown structure



Phase 1

Study the technologies

Phase 2

Collected the data

Phase 3

Train the model

Phase 4

Find the spread rate count

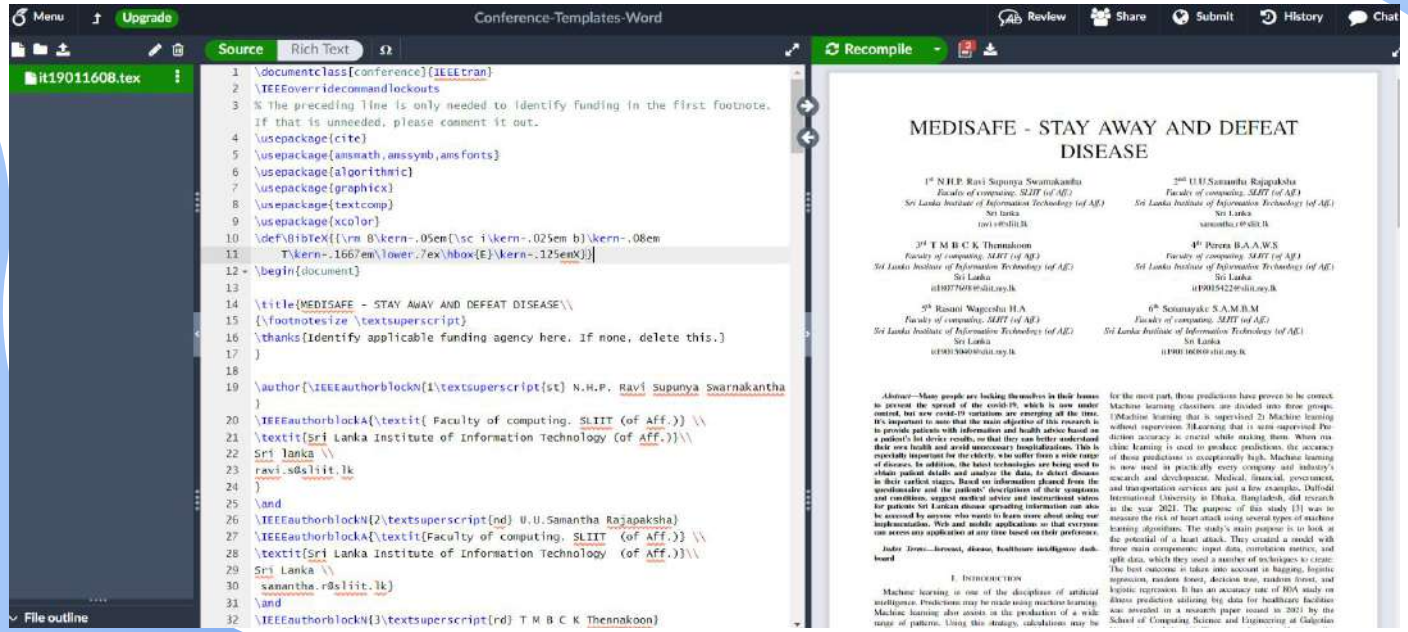
Phase 5

Display the data through the healthcare dashboard

Phase 6

Intergrade device and system , web application , mobile application.

Individual overleaf conference paper



https://www.overleaf.com/project/626ea4545e41706fc4ffa8d9

Commercialization

01

We discussed with some surgeons and they allow us to test that implemented system in their premises. (Dispensary, Hospitals)

02

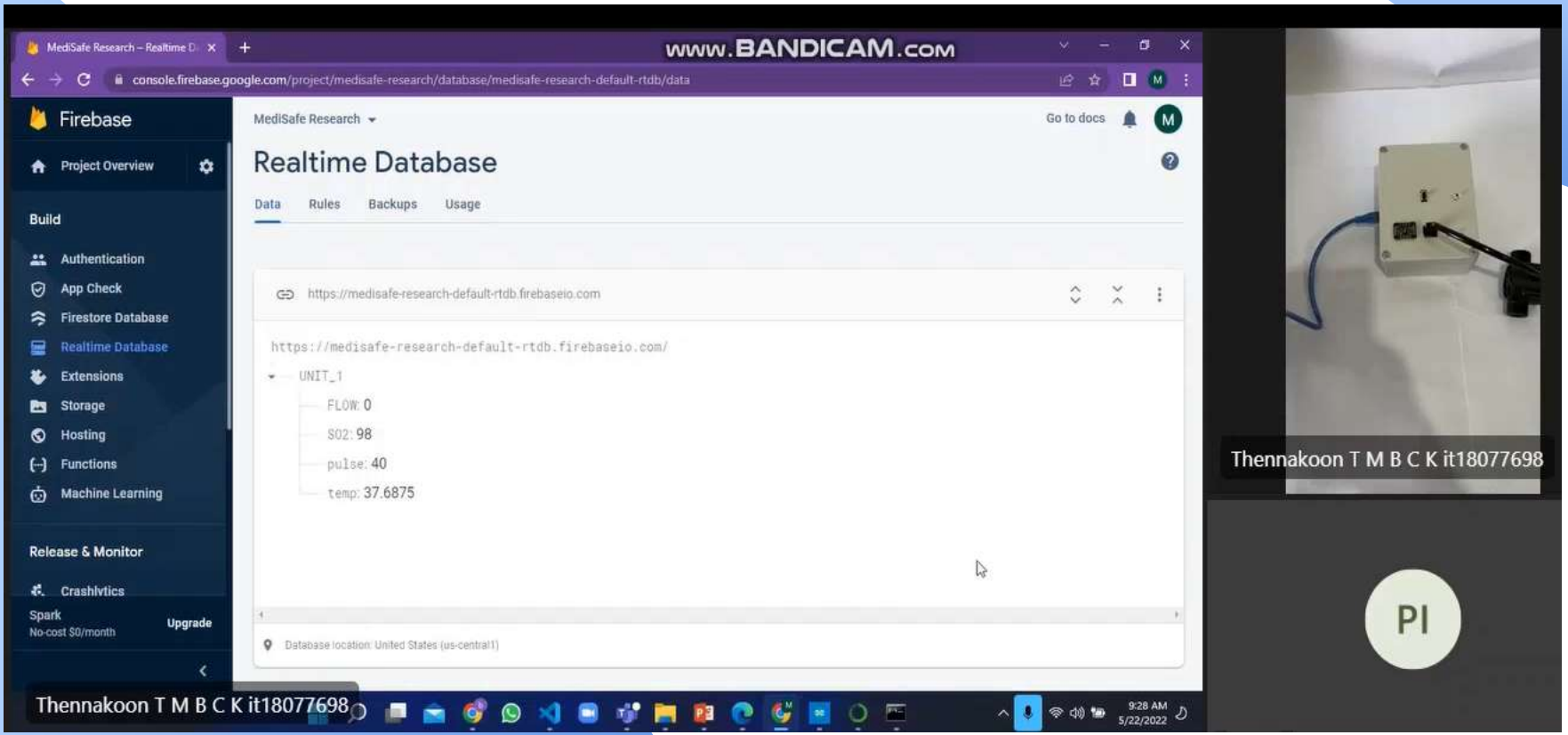
After completing our project we are hoping to introduce this to some clinical centers.

03

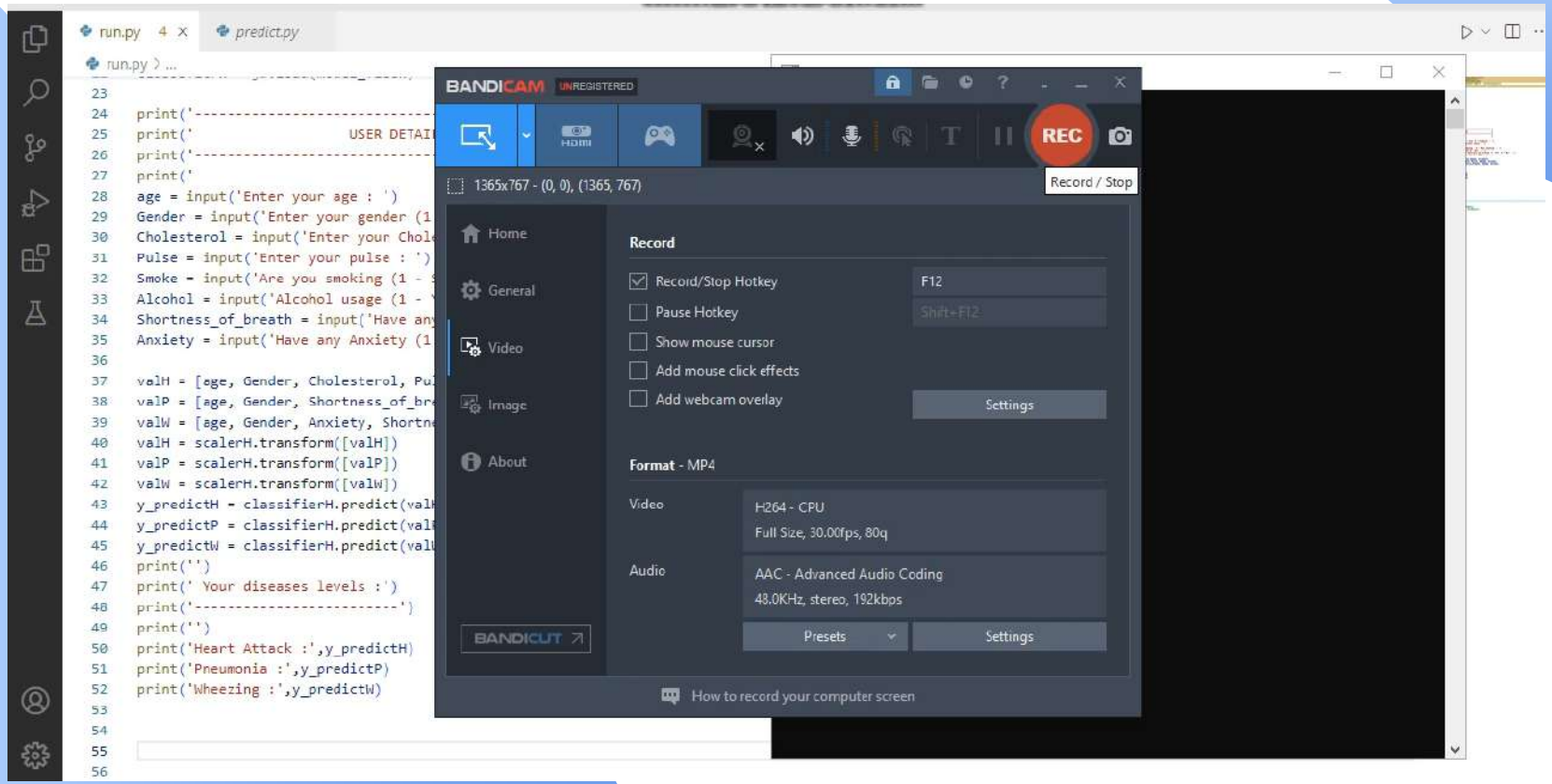
Advertise our mediSafe product via social media with its benefits.

Demonstration

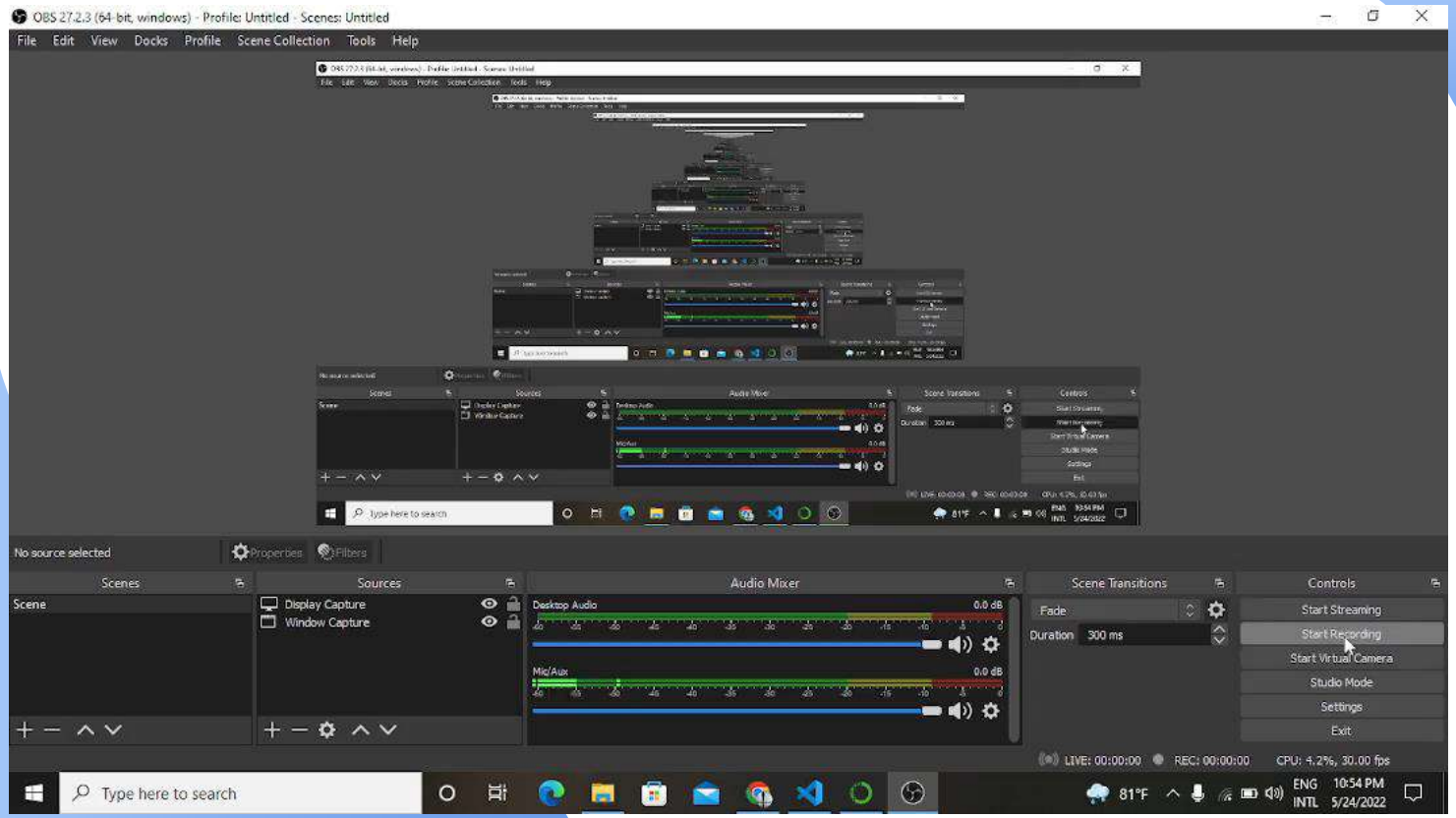
Demonstration



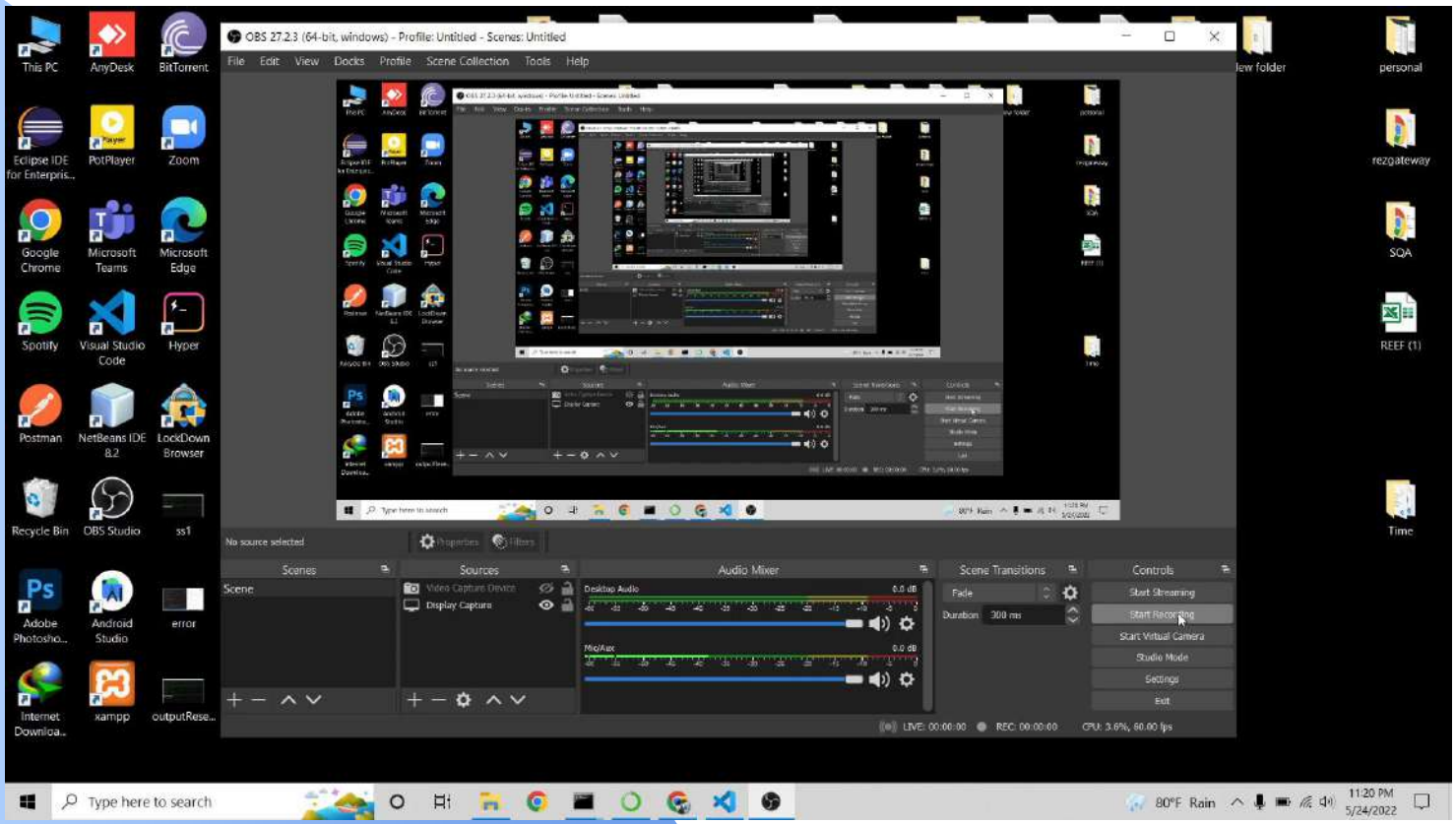
Demonstration



Demonstration



Demonstration



Thank You

Do you have any questions?