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

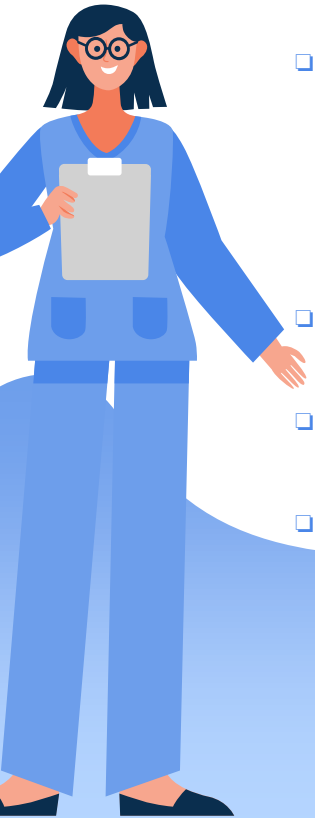
IT19011608

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 - T19, 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	11	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	1.8	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

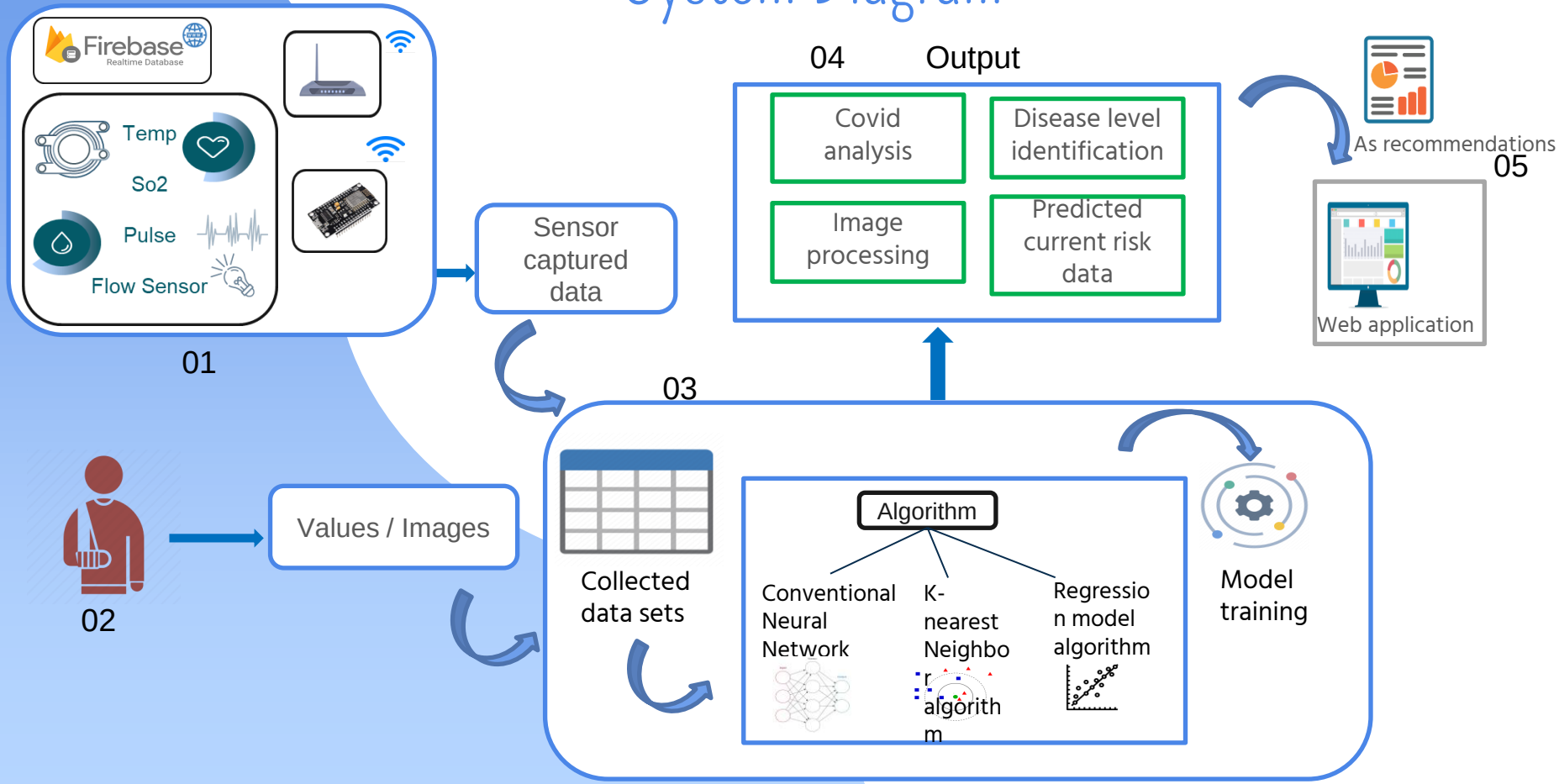
Overall solutions – 100%

- 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 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.

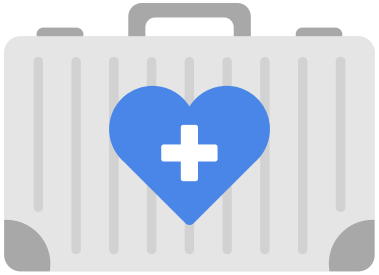
System Diagram



Focusing areas

Hospital /Medical centers

Schools



Offices

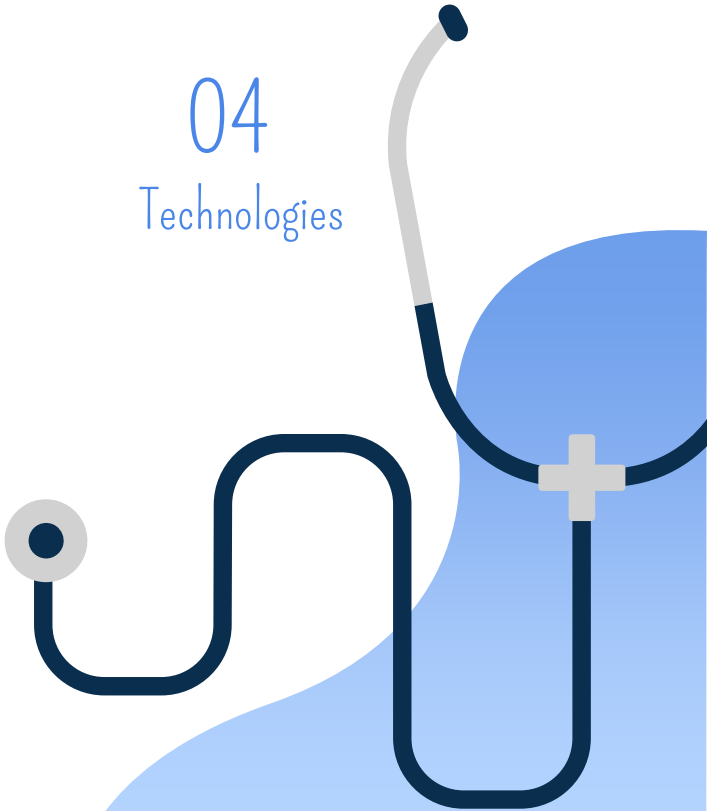
Crowded areas

IT18077698

Thennakoon T.M.B.C.K

Specialization | Information Technology

Table of Contents

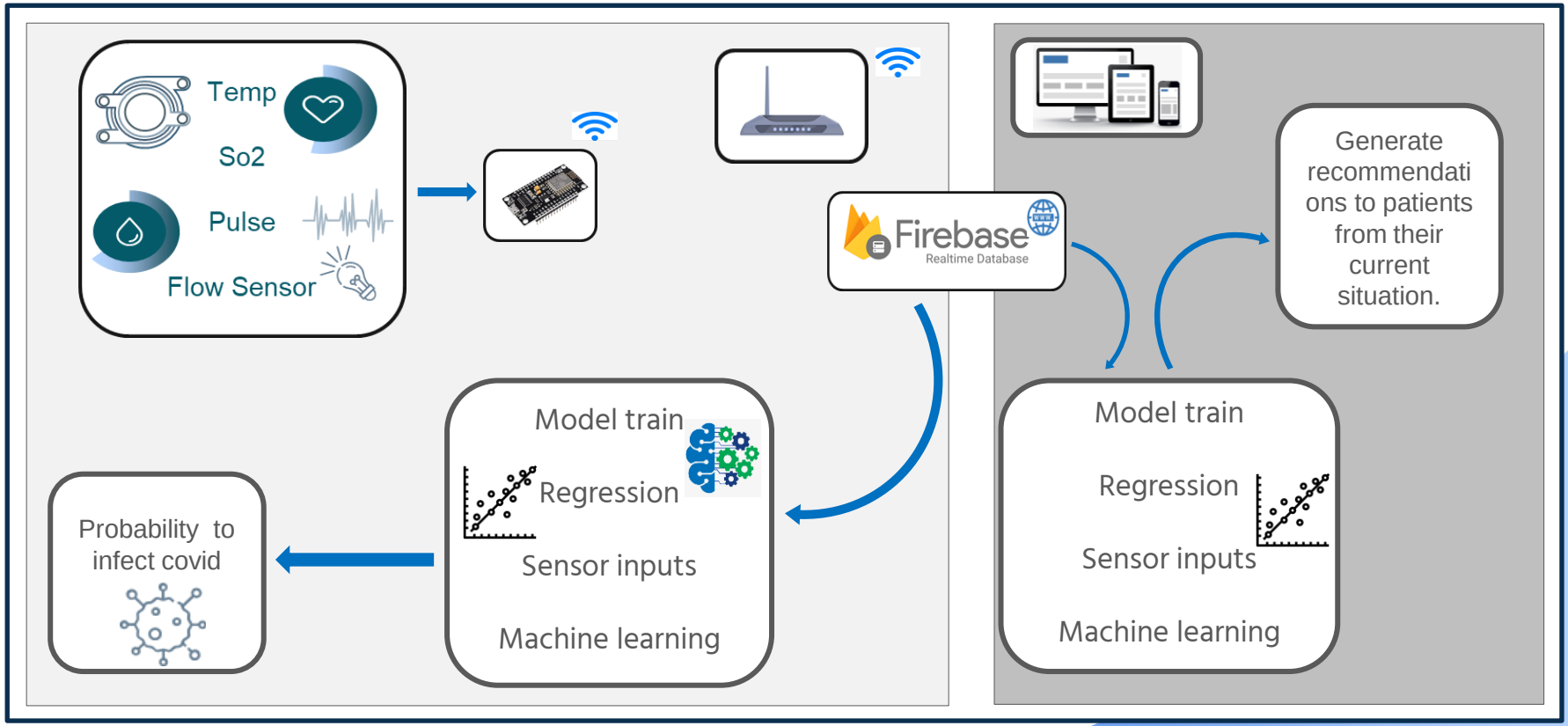


Research question

- 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.

System diagram



Progress

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

Completed(50%)

90% Progress presentation

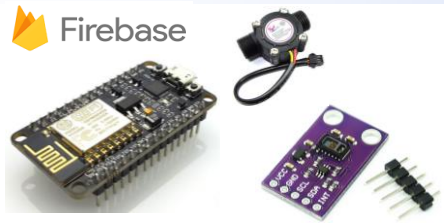
- Completed MediSafe device.
- Integrate member components together
- Generate recommendations to patients from their current situation.
- Completion of web application.

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

Final Presentation

Latest technologies in MediSafe

MediSafe device



ESP8266
Max30105sens
Flow meter
Firebase

Dataset collection and data training

Kaggle
Regression
Python / React

IDE

Jupiter Notebook
Arduino IDE
Visual studio code
Anaconda Prompt

Requirements

Functional

- Interoperability
- Accuracy
- Compliance

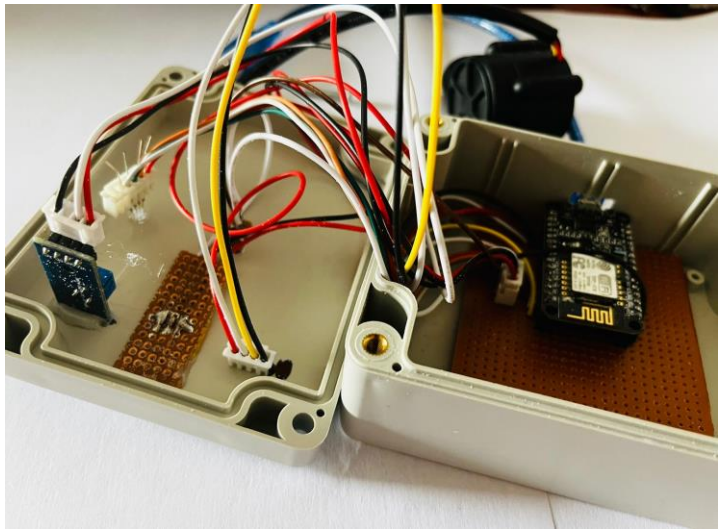
Non – functional

- Maintainability
- Manageability
- Usability
- Integrity

Risk mitigation

- This is related to medical industry research therefore the accuracy should be compulsory.
- Need to compare the actual output with a recommendation of a doctor
- Need to confirm that the output should provide the correct according to the currently available devices.

Completion of the project



Medisafe device

final2 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

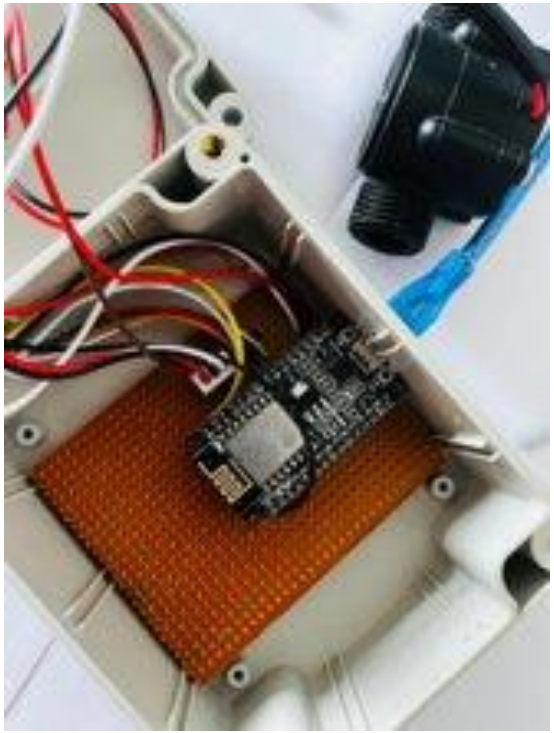
final2 § | Firebase | flow_meater | max30105 | so2 | temp

```
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 "qjnABtFp7TrCzENApcxBGQSeI21kghAo1OPwrBB5"
#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
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRA

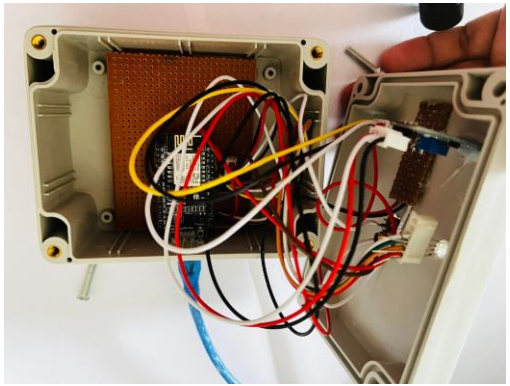
Arduino IDE

Completion of the project

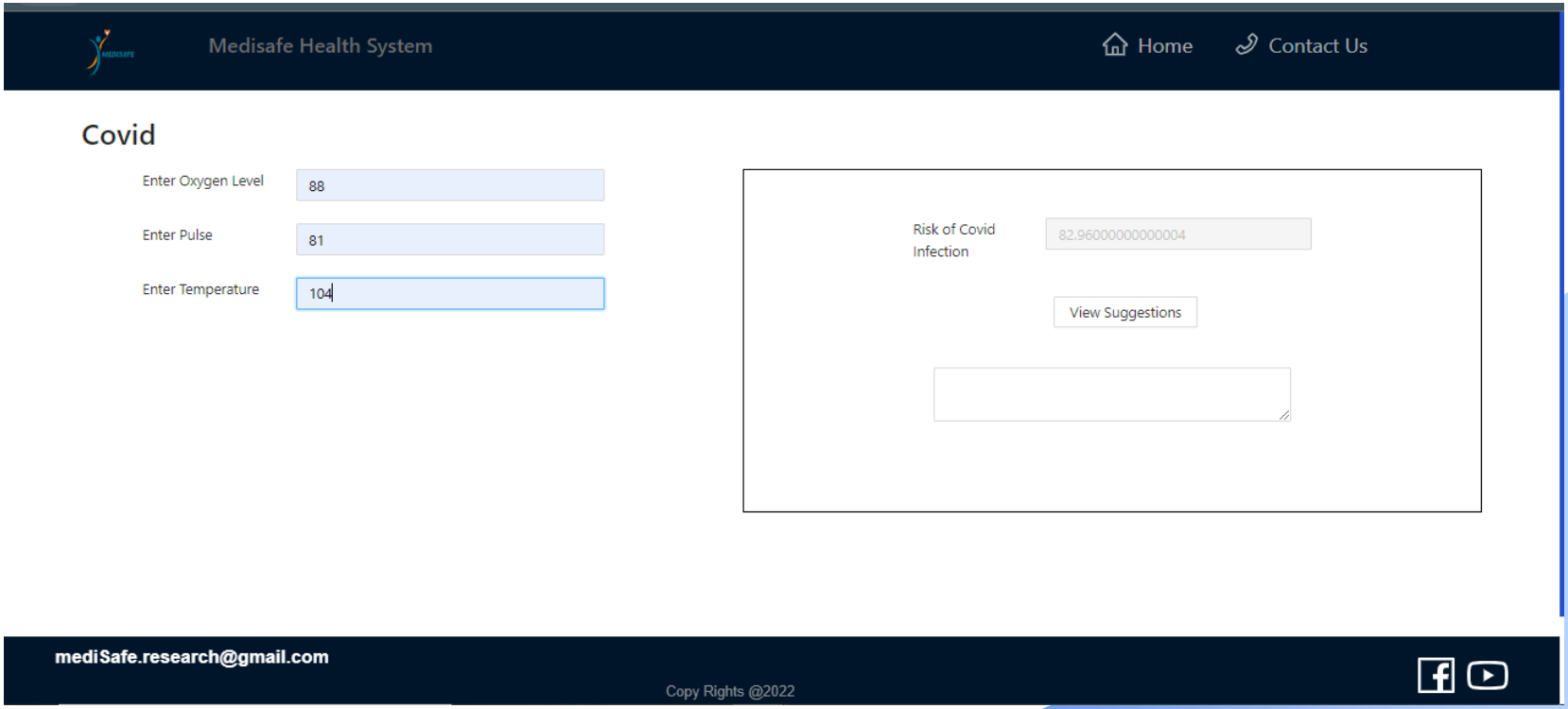


```
C:\WINDOWS\system32\cmd.exe

(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>
```

Output of prediction



Completion of the project

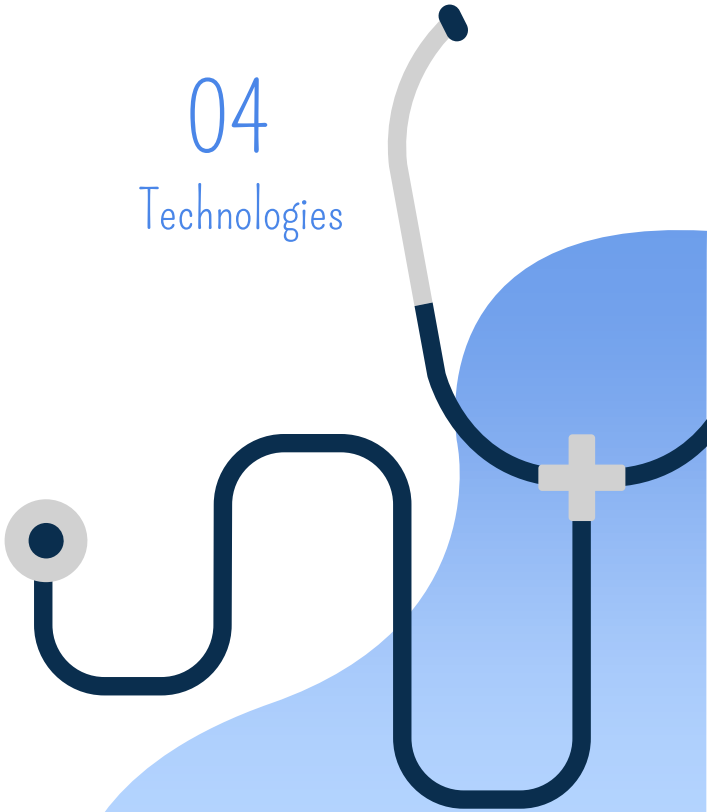


IT19015422

Perera B.A.A.W.S

Specialization | Information Technology

Table of Contents



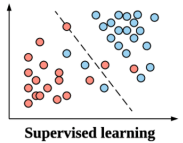
Research question

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

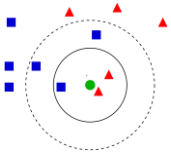
- How to provide easily recommendations via web application to the user
- What are the solutions we can give due to shortage of medicines

System diagram

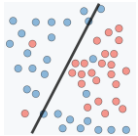
Machine learning



Supervised learning



K-nearest Neighbor algorithm



Classification model

Model training

Patient features

Train/ Test files

Disease level identification

Trained model

Gender
Pulse
Age
Anxiety
Shortness of breath
Cholesterol
Alcohol

Disease level

Web application

Suggestions/ Recommendations

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

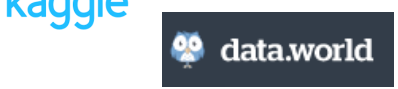
- Provide suggestions and recommendations to the user
- Web application implementation
- Integrate member components together

- Completion of web application.

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 and Language

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

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.
- ✓ Adults are not well fluent in new technologies.
- ✓ Validity of the disease level will depend on the user inputs.

Completion of the project

Frontend

1

```
<LayoutHeader />
{loading && <$Spin />}
<$Row style={{ marginTop: "100px", marginLeft: "5%" }}>
  <h1>Pneumonia</h1>
</$Row>
<$Row style={{ marginRight: "5%" }}>
  <$Col xl={12} sm={12}>
    <$Row className="jus-con-cen row-items">
      <$Col xl={5} sm={12}>
        <p>Enter Age</p>
      </$Col>
      <$Col xl={10}>
        <$Input
          name="age"
          handleChange={this.onHandleChange}
          value={form.age}
        />
      </$Col>
    </$Row>
    <$Row className="jus-con-cen row-items">
      <$Col xl={5} sm={12}>
        <p>Enter Gender</p>
      </$Col>
      <$Col xl={10}>
        <Radio.Group
          value={form.Gender}
          onChange={(e) => {
            this.onHandleChange("Gender", e.target.value);
          }}
        >
```

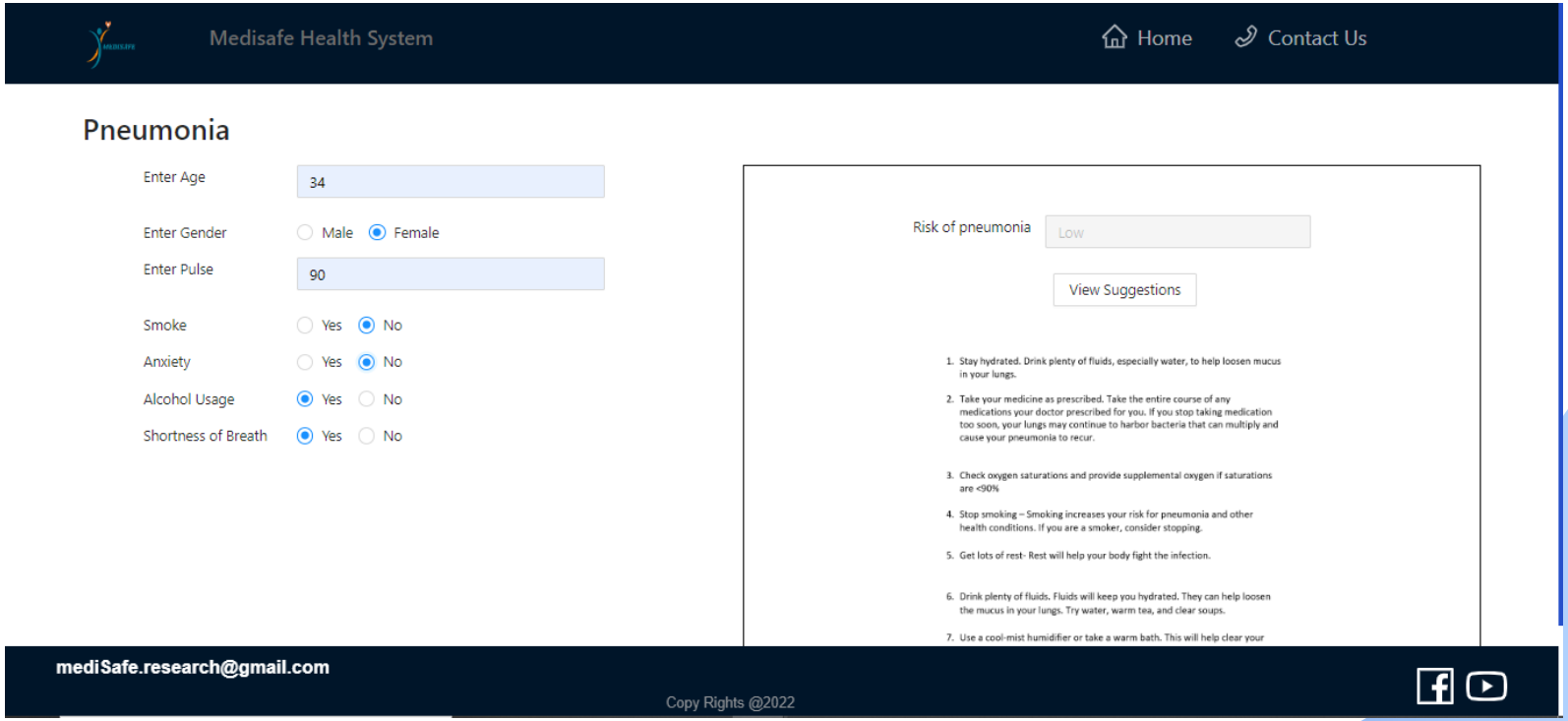
Completion of the project

API controller implementation

1

```python
@app.route('/risk', methods=['GET', 'POST'])
def predictR():
    data = {}
    post_data = request.json

    age = str(post_data['age'])
    Gender = str(post_data['Gender'])
    Cholesterol = str(post_data['Cholesterol'])
    Pulse = str(post_data['Pulse'])
    Smoke = str(post_data['Smoke'])
    Alcohol = str(post_data['Alcohol'])
    Shortness_of_breath = str(post_data['Shortness_of_breath'])
    Anxiety = str(post_data['Anxiety'])

    y_predictH, y_predictP, y_predictW = get_risk_level(
        age, Gender, Cholesterol, Pulse, Smoke, Alcohol, Shortness_of_breath, Anxiety)

    data['prediction_heart'] = y_predictH[0]
    data['prediction_wheeze'] = y_predictP[0]
    data['prediction_pneumonia'] = y_predictW[0]

    return jsonify(data)
```

Completion of the project

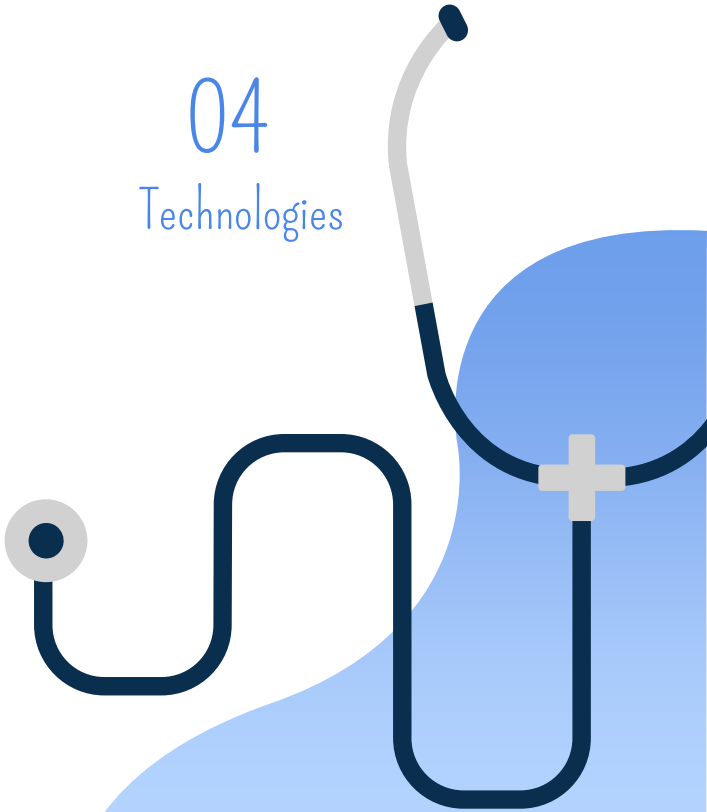


IT19015040

Rasuni Wageesha H.A

Specialization | Information Technology

Table of Contents



Research question

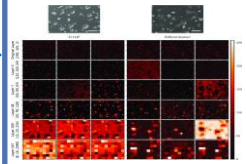
There are many different types of lung diseases and diagnosing one might be difficult.

System diagram

Image Processing

Input
Sample data

CNN



Training
Process

Training
Model

Output
Result

Input Image

Identify
disease

Web application

Provide details
of disease

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%)

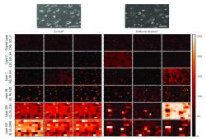
90% Progress presentation

- Provide details of the disease.
- Web application implementation.
- Integrate member components together.

- Completion of web application.
- Generate suggestion.

Final Presentation

Technologies in MediSafe

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

Requirements

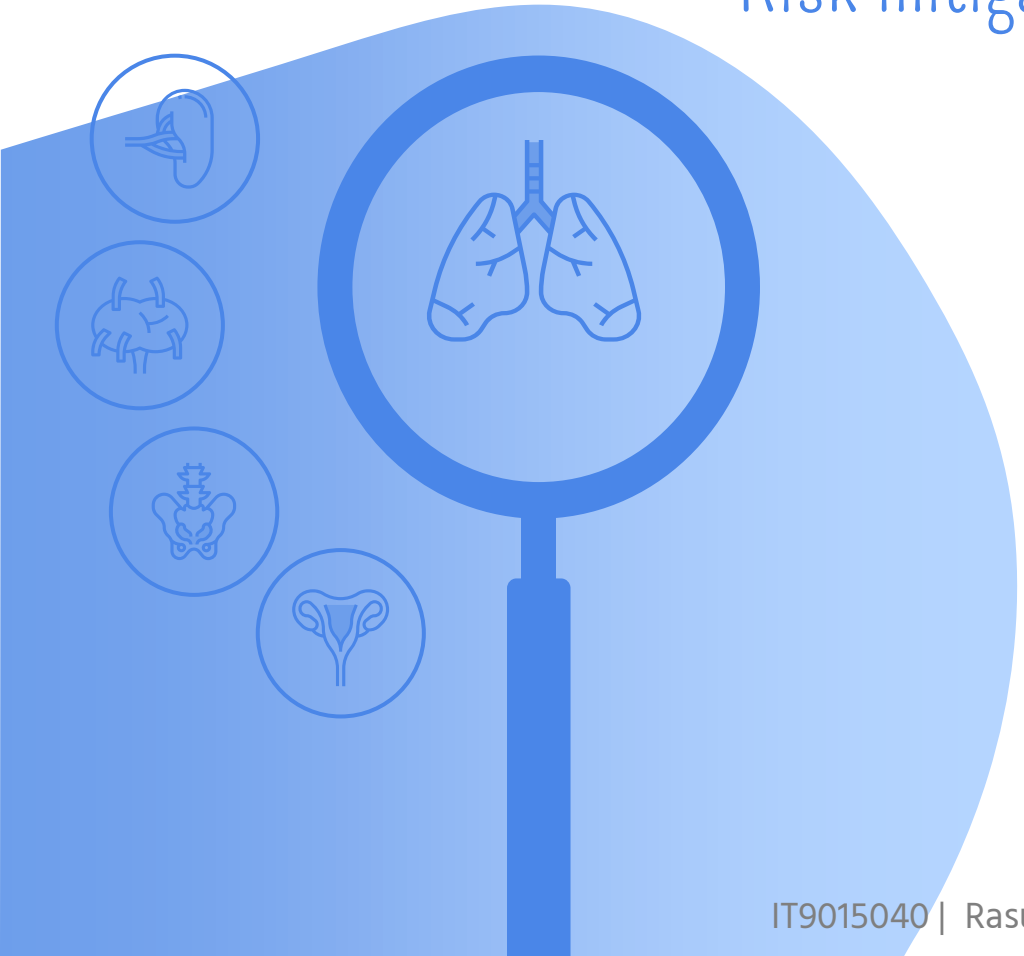
Functional

- Upload the lung image to the system.

Non – functional

- Performance
- Availability
- Reliability

Risk mitigation



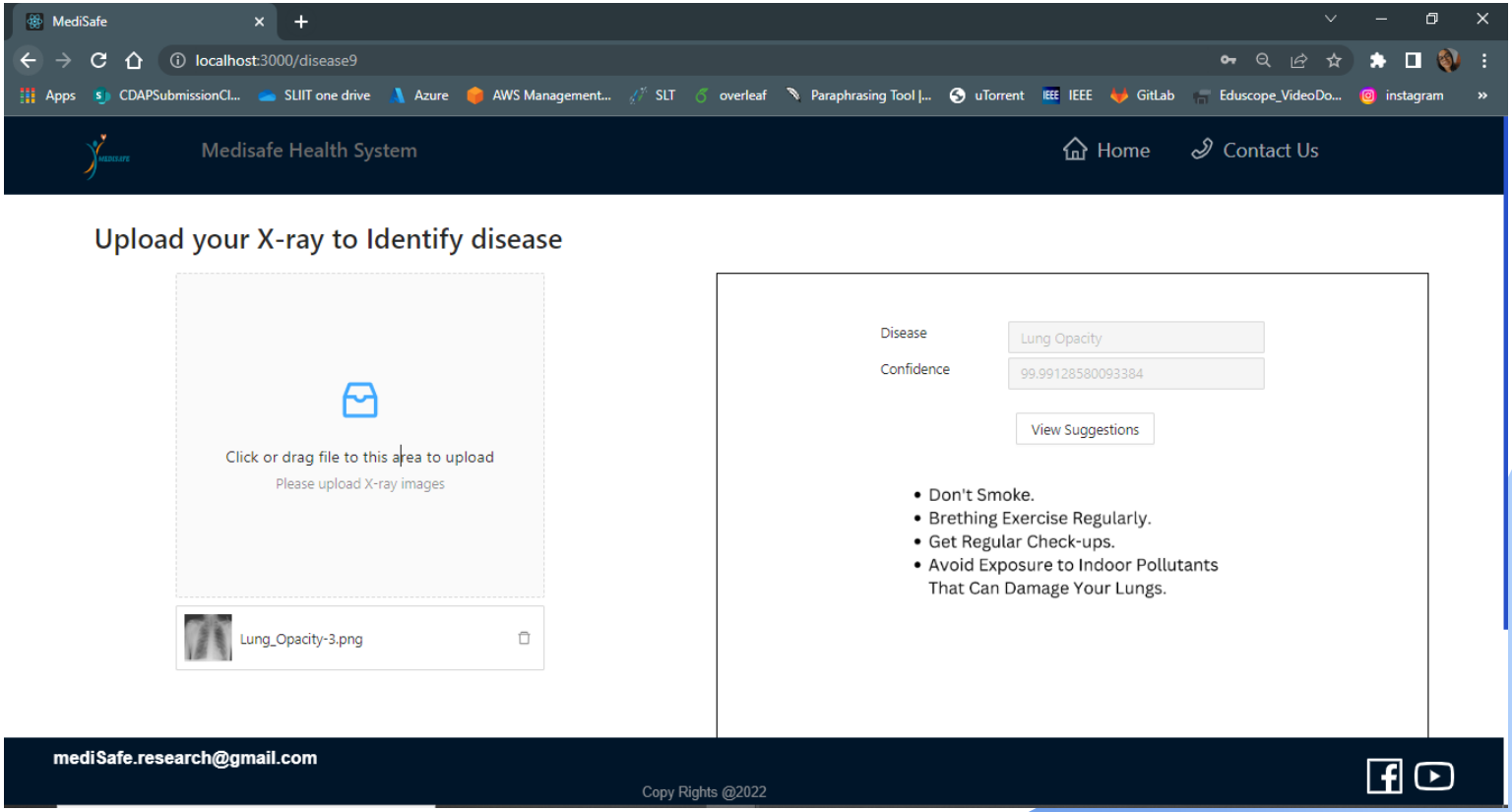
Validity of the clear images.

Completion of the project

API

```python
@app.route('/lung', methods=['GET', 'POST'])
def predict():
    data = {}

    filestr = request.files['file'].read()
    img = imgread(filestr)

    prediction_inst = []
    prediction_conf = []
    list_Of_cf = []

    outputs = model_process_img(img)
    for item in outputs['predictions']:
        list_Of_cf.append(item['confidence'])

    for item in outputs['predictions']:
        if item['confidence'] == max(list_Of_cf):
            print(item['label'], max(list_Of_cf) * 100)
            conf = max(list_Of_cf) * 100
            prediction_inst.append(item['label'])
            prediction_conf.append(conf)

    temp_val = prediction_inst[0]
    temp_conf = prediction_conf[0]
    print(temp_val)

    prediction_inst.clear()
    prediction_conf.clear()
    list_Of_cf.clear()
    data['detection'] = temp_val
    data['detectionScore'] = temp_conf

    return jsonify(data)
```

Completion of the project

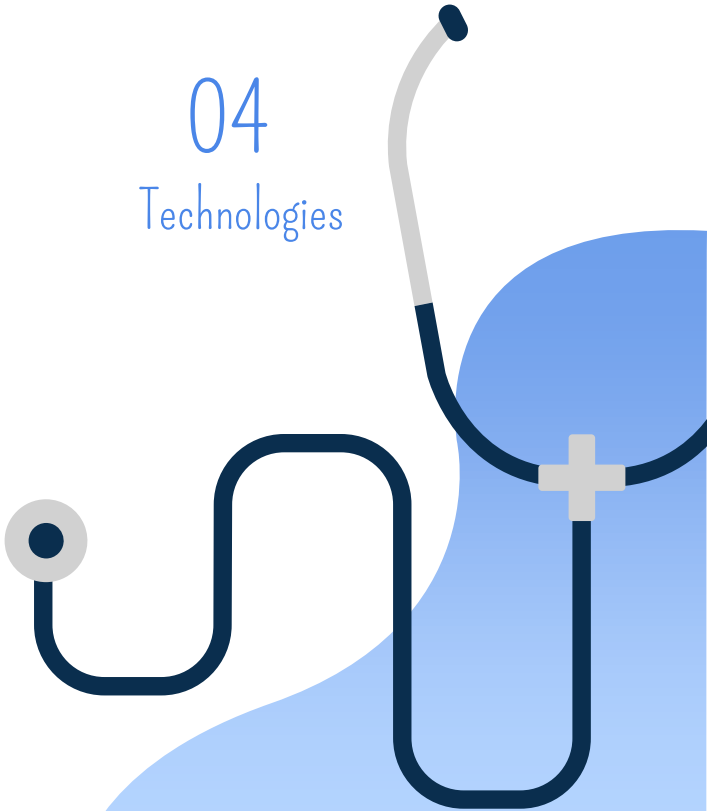


IT19011608

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

Specialization | Information Technology

Table of Contents

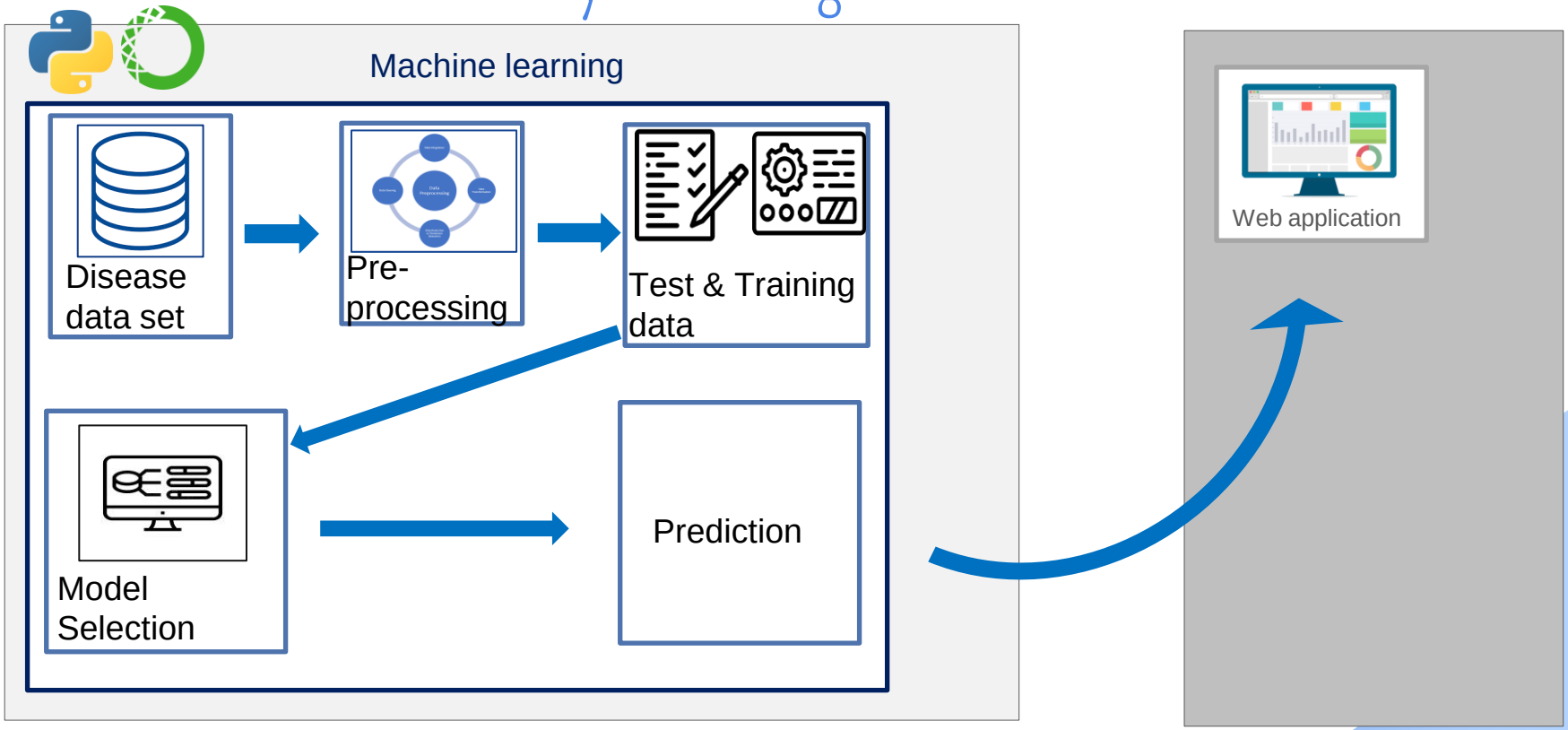


Research question

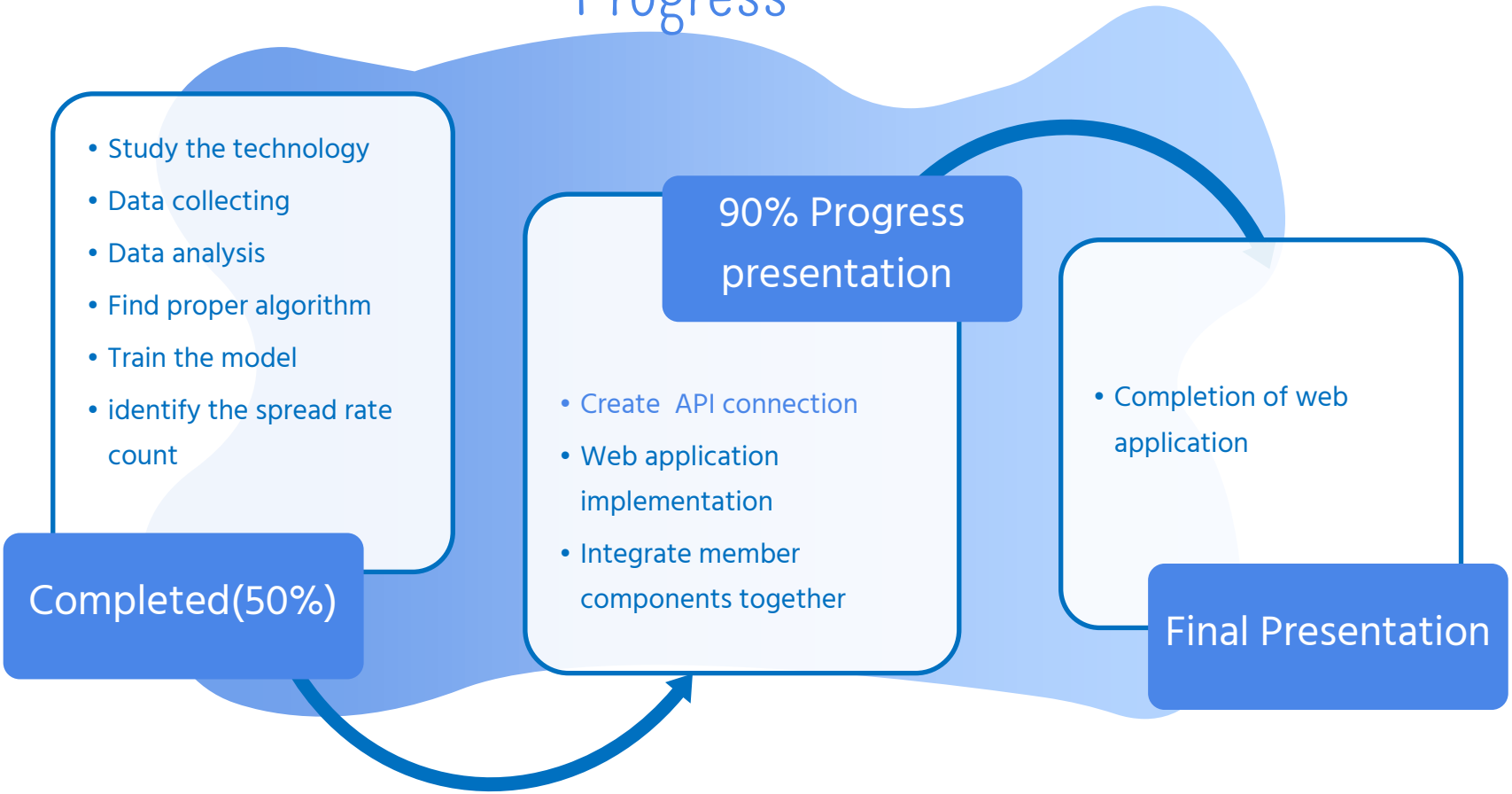
Identify the disease count on the Sri Lanka.

Target Domain

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
• Integrate member components together
• Completion of web application
Final Presentation

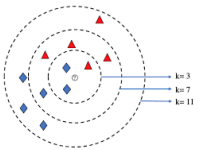
Technologies in MediSafe

Collected data

kaggle

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

Algorithem



Long short term-memory algorithms

Python / React

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

API

```python
@app.route('/spread', methods=['GET', 'POST'])
def predictS():
    data = {}
    post_data = request.json

    sickness = str(post_data['sickness'])
    city = str(post_data['city'])
    Date = post_data['Date']

    print(sickness, city, Date)

    responseArray = []
    for x in Date:
        response = get_prediction(sickness, city, str(x).split('T')[0])
        responseArray.append(response)

    data['details'] = responseArray
    return jsonify(data)
```

Completion of the project

Frontend-submit detail

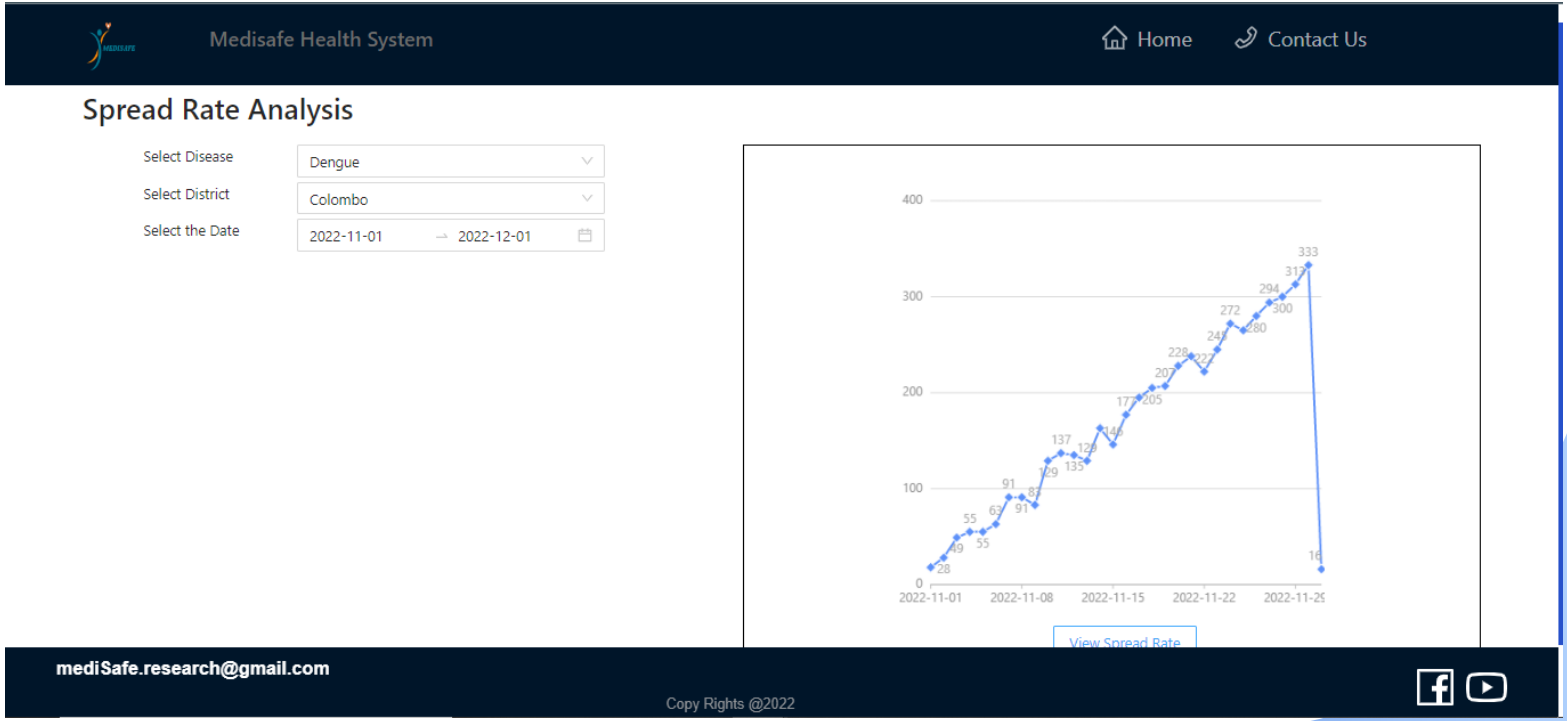
```
submit = async () => {
    const form = { ...this.state.form };
    const sickness = form.sickness;
    const city = form.city;
    let getDaysArray = this.getDaysArray(
      moment(form.Date[0]).format("YYYY-MM-DD"),
      moment(form.Date[1]).format("YYYY-MM-DD")
    );

    console.log(getDaysArray, "getDaysArray");

    const data = {
      sickness: sickness,
      city: city,
      Date: getDaysArray,
    };
    this.setState({ loading: true, getDaysArray: getDaysArray });
    try {
      await fetch("/spread", {
        method: "POST",
        headers: {
          "Content-Type": "application/json",
        },
        body: JSON.stringify(data),
      })
        .then((response) => response.json())
        .then((response) => {
          var spread = response["details"];
          this.setState({
            spread: spread,
          });
        });
      this.setState({ loading: false });
      let graph = this.getGraph(this.state.spread, this.state.getDaysArray);
      this.setState({
        graph: graph,
      });
    } catch (error) {
      this.setState({ loading: false });
    }
    this.setState({ loading: false });
  };
```

Completion of the project

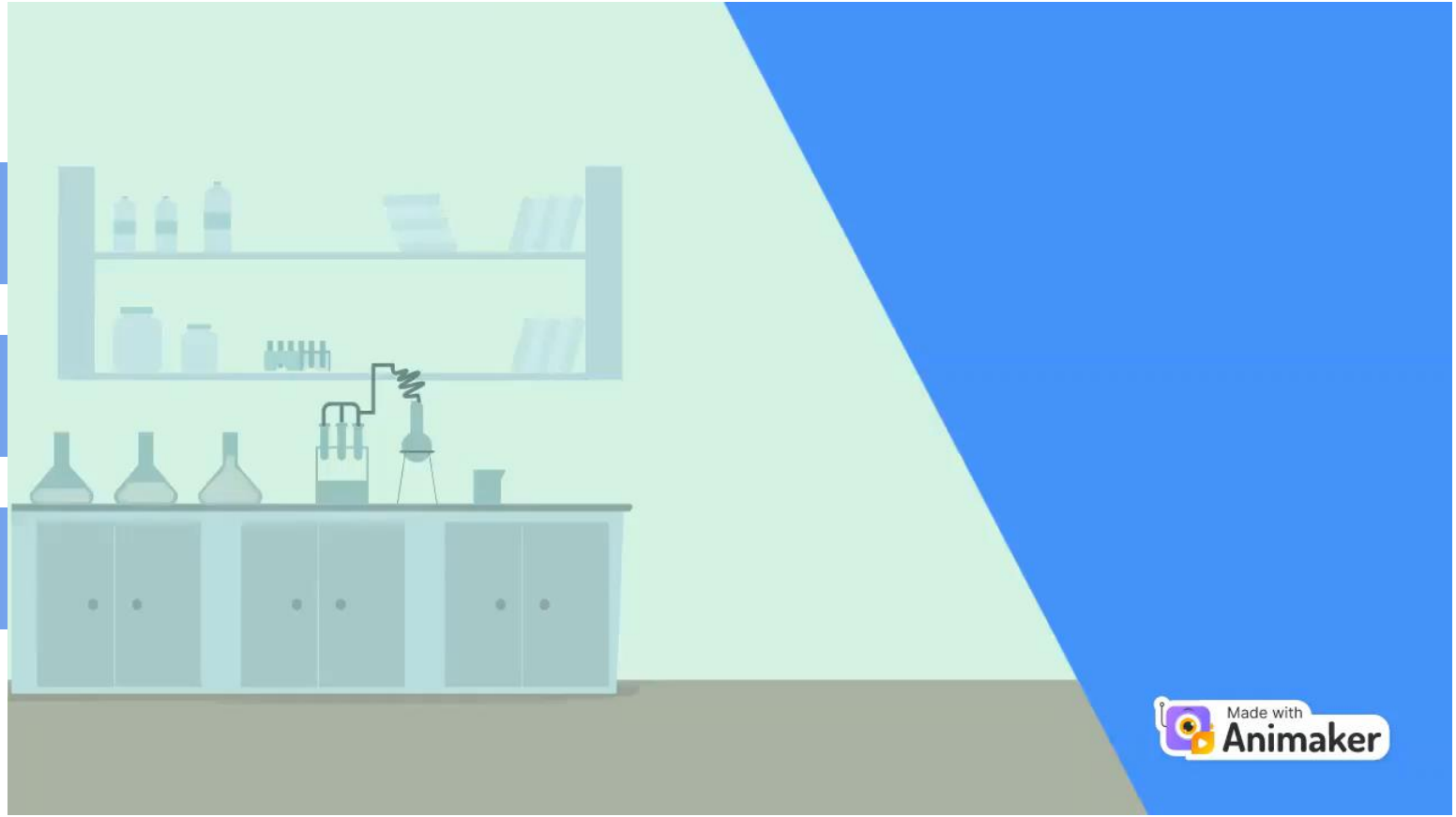
Frontend-graph

```
{graph && (
                    <Line
                      data={graph}
                      xField="Date"
                      yField="scales"
                      autoFit={false}
                      xAxis={{ range: [0, 1], tickCount: 5 }}
                      point={{ size: 5, shape: "diamond" }}
                      label={{
                        style: {
                          fill: "#aaa",
                        },
                      }}
                    />
                  )}
```

Completion of the project



Commercialization



Commercialization

01

We discussed with some surgeons, and they allow us to test that implemented system (100%) in their premises. (Dispensary, Hospitals)

02

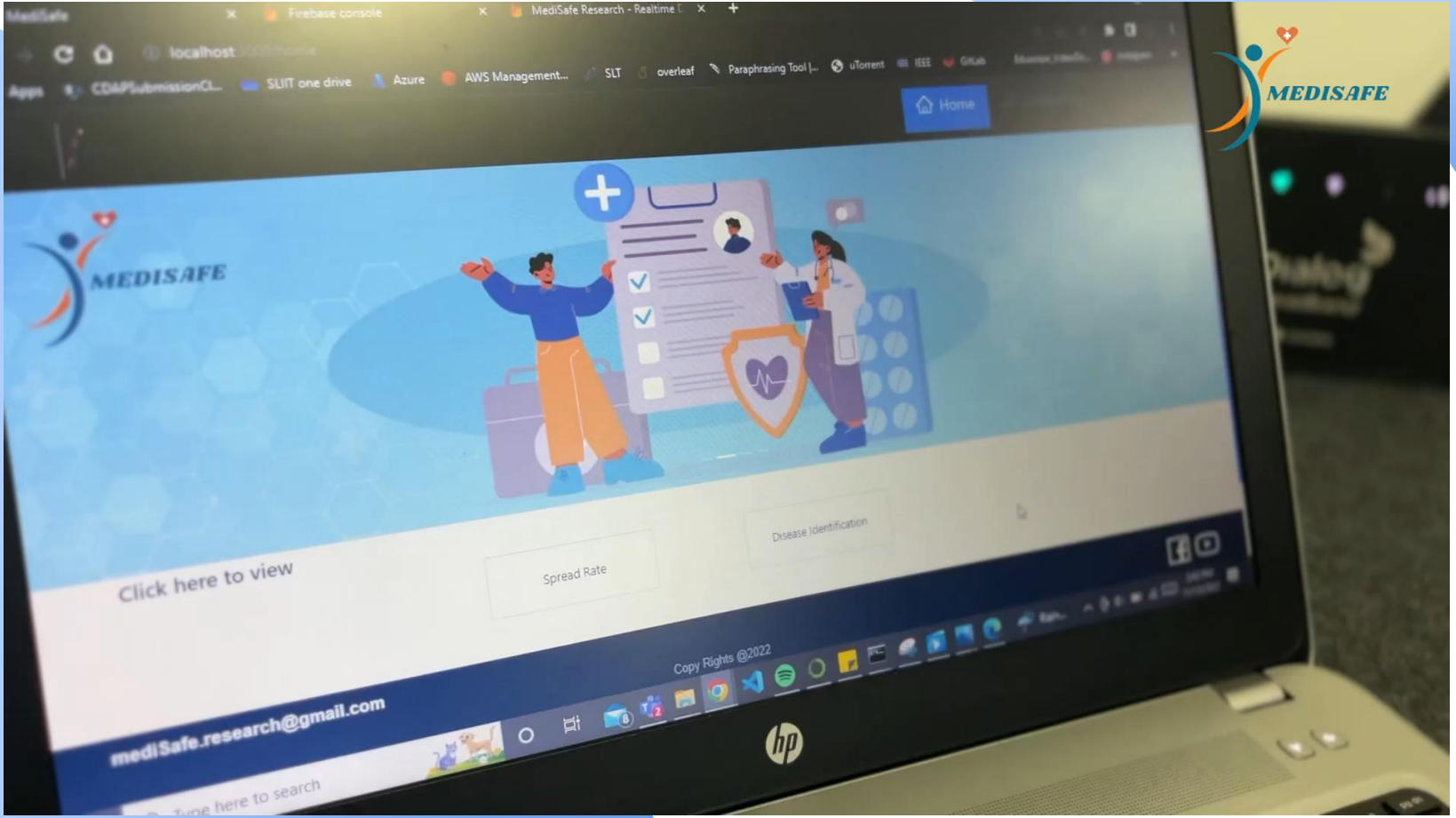
After completing (100%) our project we are hoping to introduce this to some clinical centers.

03

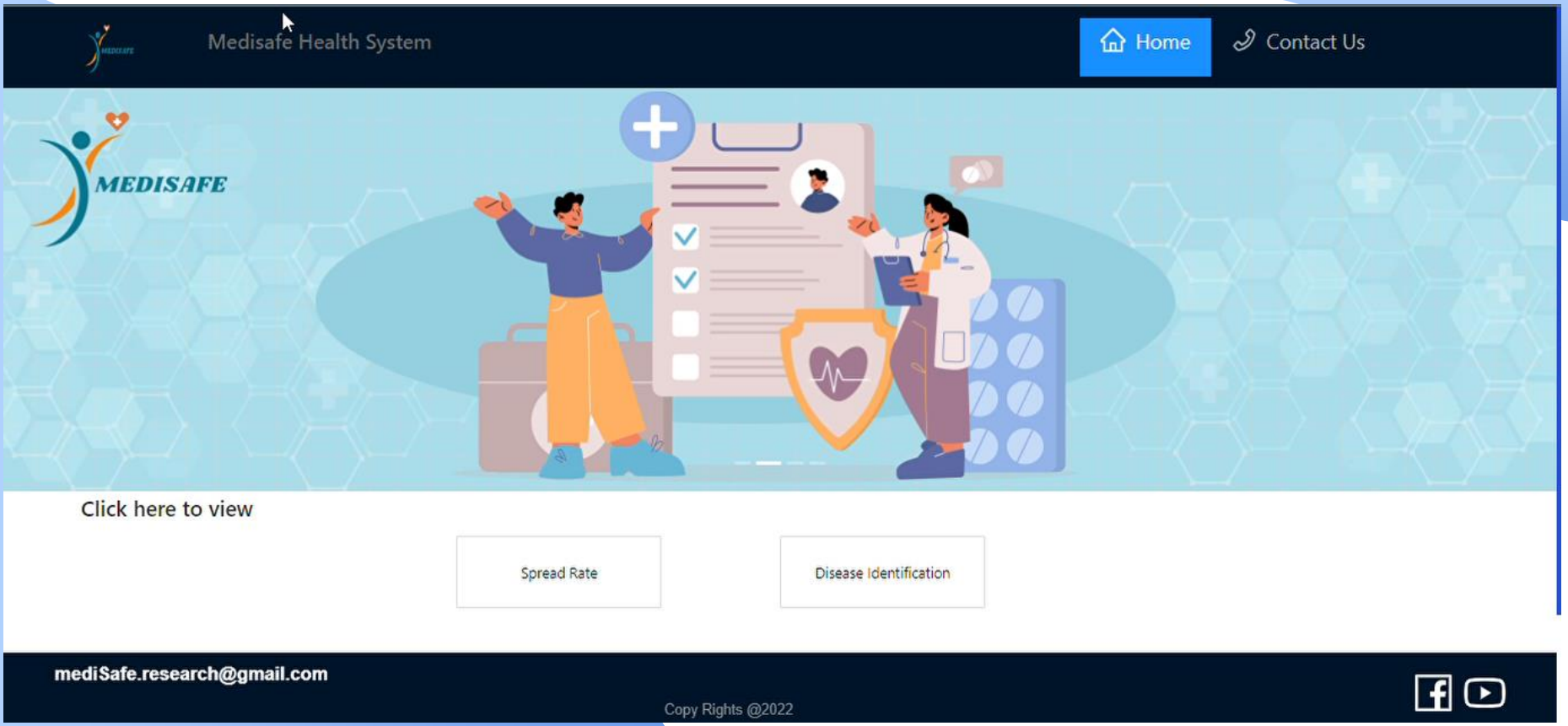
Advertise our mediSafe product via social media with its benefits.

Demonstration

Device implementation



Web app implementation



Thank You

Do you have any questions?