A Mini Project Report

COVID 19 TRACKER

Submitted in partial fulfillment for the requirements of

B.E. (CSE) V Semester Mini Project

in

COMPUTER SCIENCE AND ENGINEERING

by

Roll No.	Student Name	Mentor Name
Somisetty Akshay	160119733025	Sri. B Ramadasu sir
S Rohit	160119733043	Sri. B Ramadasu sir

Department of Computer Science and Engineering,

Chaitanya Bharathi Institute of Technology (Autonomous),

(Affiliated to Osmania University, Hyderabad)

Hyderabad, TELANGANA (INDIA) – 500 075

November-2021

Annexure-2

CERTIFICATE

This is to certify that the project titled "**COVID TRACKER**" is the bonafide work carried out by **Rohit(160119733043) and Akshay(160119733025)** students of B.E.(CSE) of Chaitanya Bharathi Institute of Technology, Hyderabad, affiliated to Osmania University, Hyderabad, Telangana(India) during the academic year 2020-2021, submitted in partial fulfillment of the requirements for the **B.E.(CSE) V Semester Mini Projects** and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Mentor(s)
Sri. B Ramadasu sir

Head, CSE Dept.
Smt. Y. Ramadevi,
Professor

DECLARATION

We hereby declare that the project entitled "**COVID 19 TRACKER**" submitted for the B. E (CSE) Mini Projects is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Name(s) and Signature(s) of the Student

Place: Hyderabad **Sunkara Rohit**

Date: 17-11-2021 **Somisetty Akshay**

ABSTRACT

COVID-19 declared as a global pandemic by WHO, has emerged as the most aggressive disease, impacting more than 90% countries of the world.

The virus started from a single human being in China, is now increasing globally at a rate of 3% to 5% daily and has become a never ending process. Some studies even predict that the virus will stay with us forever. India being the second most populous country of the world, is also not saved, and the virus is spreading as a community level transmitter.

Therefore, it becomes really important to analyse the possible impact of COVID-19 in India and forecast how it will behave in the days to come. Thus a lot of prediction models are coming up amidst this pandemic. This is our attempt at making a model with simple linear regression without considering other complex parameters(Epidemiology, Socio-Cultural parameters). These predictions might be helpful for the time series projection of COVID-19 across the world.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

We would like to express our sincere gratitude and indebtedness to our project guides and mentors Sri. B Ramadasu sir who has supported us throughout our project with patience and knowledge and guided us with valuable inputs and suggestions.

We are also thankful to our Principal and Head of the department Dr. Y.Rama Devi, for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Rohit

Akshay

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

PROBLEM DEFINITION

COVID-19 declared as a global pandemic by WHO, has emerged as the most aggressive disease, impacting more than 90% countries of the world. The virus started from a single human being in China, is now increasing globally at a rate of 3% to 5% daily and has become a never ending process. Some studies even predict that the virus will stay with us forever. India being the second most populous country of the world, is also not saved, and the virus is spreading as a community level transmitter. Therefore, it becomes really important to analyse the possible impact of COVID-19 in India and forecast how it will behave in the days to come. Thus a lot of prediction models are coming up amidst this pandemic. This is our attempt at making a model with simple linear regression without considering other complex parameters(Epidemiology, Socio-Cultural parameters). These predictions might be helpful for the time series projection of COVID-19 across the world.

METHODOLOGIES

To make developments more frequent better data sets have to be developed. They will help us make better predictions about the pandemic. Packages like pandas, numpy and scikit learn will help in this process.

SCOPE OF THE PROJECT

The Covid 19 tracker which we have developed may be used by concerned health authorities to control the pandemic.

ORGANIZATION OF THE REPORT

The report submitted is of 6 chapters where each chapter has several sub topics. Chapter 1 explains about the problem and its general solution and the next chapter compares the present solution and the proposed solution. Chapter 3 explains the algorithm or the flow of the proposed solution. Chapter 4 provides the implementation part of the proposed system. Chapter 5 discusses the results of the proposed solution. The last chapter concludes by describing a few limitations of the project.

CHAPTER 2

LITERATURE SURVEY

Introduction to the problem domain terminology

Owing to the erratic nature of the virus, it becomes really important to analyse the possible impact of COVID-19 in India and forecast how it will behave in the days to come. This is our attempt at making a model with simple linear regression.

Existing Systems

As the pandemic has taken us by surprise, there are no prediction systems presently. Most of the trackers and predictors were developed after March when the effect of the virus was observed on a global scale.

Tools/ Technologies Used

Software requirements

Processor – intel Pentium core i5

OS – Windows 10

Languages – Python, HTML, CSS, JavaScript

Frameworks – Materialize CSS, jQuery, scikit learn, pandas, numpy, Flask.

Hardware requirements

Monitor Resolution 1024 X 768 or higher

Processor(CPU) with 2 gigahertz (GHz) frequency or above

Internet Connection Broadband (high-speed)

A minimum of 2GB RAM or more

CHAPTER 3

DESIGN OF THE PROPOSED SYSTEM

BLOCK DIAGRAM

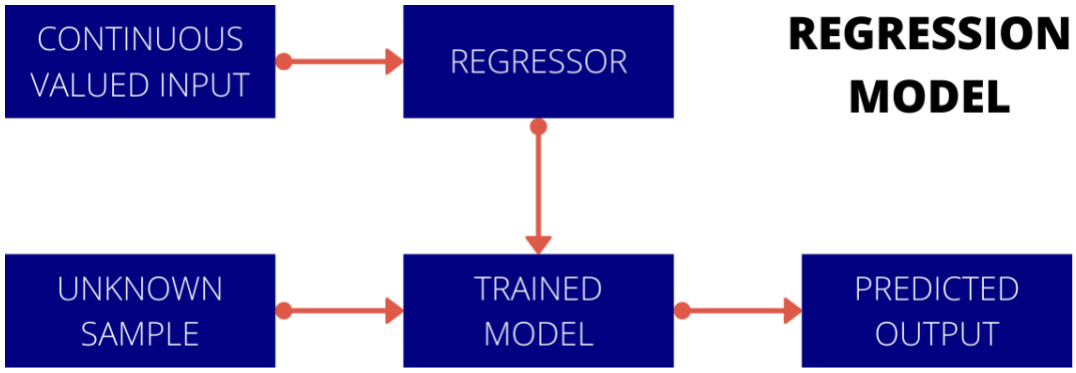


FIG 3.1: BLOCK DIAGRAM OF THE PROPOSED ML MODEL

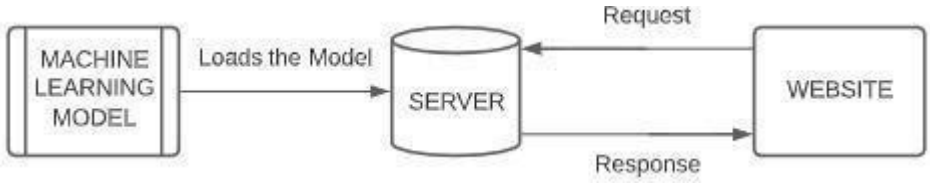


FIG 3.2: BLOCK DIAGRAM OF THE ENTIRE WORKFLOW OF OUR PROJECT

MODULE DESCRIPTION

FRONTEND

The frontend is kept very simple. The user is directly taken to the home page with a **“GET PREDICTIONS”** button. On clicking that button the user is redirected to the predictions page where they get to choose a country and date out of a drop box.

By pressing the **“GET PREDICTION”** button the predictions for the given country and date are displayed.

SERVER

The program used is in Python 3.8 version, the web framework used is flask. The server loads the machine learning models which are stored in a pickle file. When the server receives a request with location and date. It sends the data to the model and gets the prediction, this prediction is sent as a response.

MACHINE LEARNING MODEL

The prediction model is implemented using various python frameworks scikit learn, numpy and pandas.

CHAPTER 4

IMPLEMENTATION OF THE PROPOSED SYSTEM

FLOWCHART

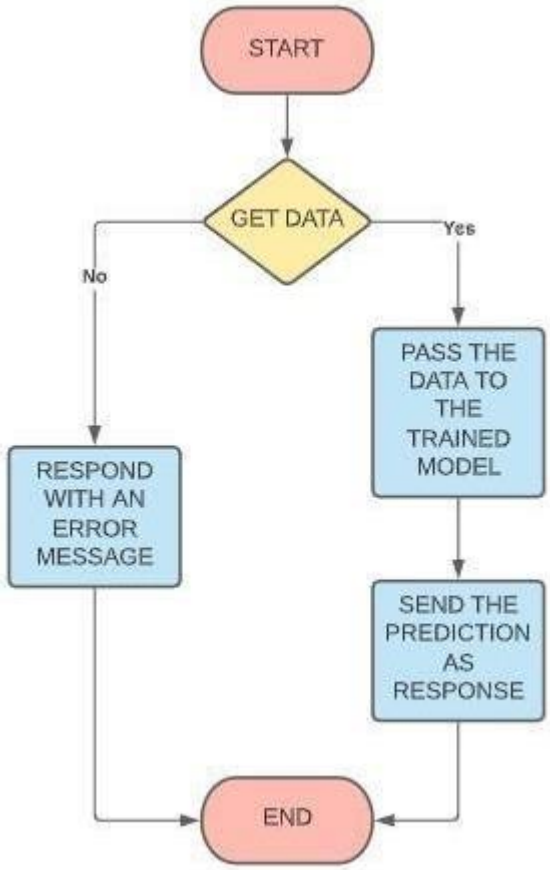


FIG 4.1: FLOW CHART OF THE SERVER WORKFLOW

Here in the flowchart we can see that the server gets the data from the user. This is implemented using the **"GET PREDICTIONS"** button.

The functionalities offered to the user are:

1. Selection of a country.
2. Selection of date.

DESIGN AND TEST STEPS

Python 3.8 version is used, the web framework used is flask. The prediction model is implemented using various python frameworks like scikit learn, numpy and pandas.

WEB DEVELOPMENT

When the HTML document is loaded, the server starts and loads the models and responds with the locations.

The user will be prompted with a get predictions button. On clicking that he will be provided with a form containing location and date picker. On submitting the form, the data is sent to the server.

The server sends the data to the model and receives the prediction. This prediction is sent as a response.

This response is shown to the user.

MACHINE LEARNING MODEL

The data gathered is first preprocessed. Now, we have made an 80:20 train-test split on the data. The train-data is used to train our model, test-data is unknown to the model. So, this will be helpful in knowing the performance of the trained model. Then we train a model based on the data.

ALGORITHMS / PSEUDO CODE

Frontend

The front-end of the project is the user interface. We made use of HTML, CSS, JS and Materialize CSS, jQuery frameworks for the frontend.

The initial homepage and further modals are displayed using this file.

The user is provided with a button to get the predictions. On clicking the button he will be prompted with a form containing location and a date picker. After submitting the form, the form data is sent to the flask server. The server responds with the prediction. Which is displayed to the user.

```
<html lang="en">

<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Covid Cases Predictor</title>

    <!-- CSS -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize
.min.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <style>
        body {
            margin: 0 !important;
            padding: 10vh 10vw;
            min-height: 100vh;
            min-width: 100vw;
            background-image: url('./home.jpg');
            background-position: center;
            background-repeat: no-repeat;
            background-size: cover;
        }

        #trigger {
            margin: auto;
            margin-bottom: 0;
        }

        input {
            margin: 0px !important;
        }

        form>.row {
            margin: 30px;
        }

        div .btn-large {
            width: 100%;
            font-weight: 500;
        }

        .select-wrapper {
            width: 100%;
        }
```

```
        .select-dropdown {
            max-height: 250px;
        }
    </style>

    <!-- JavaScript -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scrip
t>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.m
in.js"></script>
</head>

<body class="valign-wrapper row" onload="ready()">
    <button data-target="form" class="btn-large modal-trigger blue"
id="trigger">Get  Predictions</button>
    <div id="form" class="modal">
        <div  class="modal-content">
            <button class="btn-floating modal-close waves-effect waves-light
right  red"><i
                    class="material-icons">close</i></button>
            <form class="col s12">
                <div class="row valign-wrapper">
                    <div class="col s12 m6 l4">
                        <label for="location" class="btn-large waves-effect
waves-light indigo">Location</label>
                    </div>
                    <div class="col s12 m6 l8 valign-wrapper">
                        <select id="location">
                            <option value="" disabled selected>Choose
location</option>
                        </select>
                    </div>
                </div>
                <div class="row valign-wrapper">
                    <div class="col s12 m6 l4">
                        <label for="date" class="btn-large waves-effect
waves-light indigo">Date</label>
                    </div>
                    <div class="col s12 m6 l8 valign-wrapper">
```

```
                        <input type="text" id="date" class="datepicker">
                    </div>
                </div>
                <div class="row center-align">
                    <div class="col s12">
                        <button class="btn-large waves-effect waves-light
green modal-trigger" data-target="result" type="submit" style="width: auto;"
onclick="predict(event)">Get prediction</button>
                    </div>
                </div>
            </form>
        </div>
    </div>

    <div id="result" class="modal bottom-sheet">
        <div  class="modal-content">
            <div style="display: flex; justify-content: space-between;">
                <h3 class="header">Our Prediction</h3>
                <button class="btn-floating modal-close waves-effect
waves-light  right  red"><i
                        class="material-icons">close</i></button>
            </div>
            <div class="center" id="input" style="width: 60%; margin: 0 auto;
font-size: 20px; display: flex; justify-content: space-between;">
            </div>
            <table class="centered responsive-table">
                <thead>
                    <tr>
                        <th>Total Cases per million</th>
                        <th>Total Deaths per million</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td  id="total_cases"></td>
                        <td  id="total_deaths"></td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
```

```
    <script>
        let data;
        const ready = () => {
            M.Range.init(document.querySelectorAll('input[type=range]'));
            M.Datepicker.init(document.querySelectorAll('.datepicker'), {
                format: 'dd-mm-yyyy'
            });
            M.Modal.init(document.querySelectorAll('.modal'));

            $.get("http://127.0.0.1:5000/get_locations", (response, status) =>
{
                if (response) {
                    data = response.locations;
                    let locations = "",
                        location;
                    data.forEach(location => {
                        locations += `<option
value="${location}">${location}</option>`;
                    });
                    document.querySelector('#location').innerHTML +=
locations;

                    M.FormSelect.init(document.querySelector('#location'));
                }
            });
        }

        const predict = e => {
            e.preventDefault();
            let date = document.querySelector('#date').value;
            $.post("http://127.0.0.1:5000/predict", {
                    location: document.querySelector('#location').value,
                    year: parseInt(date.substring(6)),
                    month: parseInt(date.substring(3, 5)),
                    day: parseInt(date.substring(0, 2))
                },
                (response, status) => {
                    document.querySelector('#result .modal-content
#input').innerHTML = `
                    <p>Location:
${document.querySelector('#location').value}</p>
                    <p>Date: ${document.querySelector('#date').value}</p>
```

```
                `;
                    document.querySelector('#result .modal-content
#total_cases').innerHTML = response.prediction.total_cases;
                    document.querySelector('#result .modal-content
#total_deaths').innerHTML = response.prediction.total_deaths;
                });
            M.Modal.getInstance(document.querySelector('#result')).open();
        }
    </script>
</body>

</html>
```

Machine Learning Model

The Machine learning model uses linear regression to extrapolate the data and make future predictions.

It can also be used as a covid 19 tracker as it has already registered the previous data. It can be used to conduct studies on the flow of the virus during its initial stages since November 2019.

7 steps of machine learning:

1. Gathering Data

   We have gathered the data from https://ourworldindata.org/

2. Preparing that data

   This is an important step in the machine learning process. Is this step we should clean the data and make the necessary modifications to data. Outliers should be handled carefully. This step is also known as Data Preprocessing.

3. Choosing a model

   We have chosen a linear regression model.

4. Training

   Then we need to train our data with the chosen model. We have made a 80:20 train-test split so that our model gets trained well and we will have 20% of the data as unknown, So we can know how our model is working

5. Evaluation

   In this step we will use the test data(20% of original data) for the evaluation process.

6. Hyperparameter Tuning

7. Prediction

   In this step we will make assumptions on the predictions and come to a conclusion of the performance of the trained model.

```
#!/usr/bin/env  python
# coding: utf-8

# # IMPORTING NECESSARY PACKAGES

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

get_ipython().run_line_magic('matplotlib', 'inline')


# In[2]:


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression


# In[3]:


import pickle
import json
```

```
# # READING CSV FILE

# In[4]:


df = pd.read_csv("data.csv")
df.head()


# # DATA PREPROCESSING

# In[5]:


df['year'] = df.date.str.slice(start=6)
df['month'] = df.date.str.slice(start=3, stop=5)
df['day'] = df.date.str.slice(start=0, stop=2)
df.head()


# In[6]:


df.isnull().sum()


# In[7]:


df = df.fillna(0)


# In[8]:


df = pd.concat([df, pd.get_dummies(df.location)], axis = 'columns')
df.head()


# In[9]:
```

```
df = df.drop(['date', 'location'], axis = 'columns')
df.head()


#  In[10]:


X = df.drop(['total_cases_per_million', 'total_deaths_per_million'], axis =
'columns')
X.head()


# In[11]:


data = {
    'columns': [col for col in X]
}
with open('data.json', 'w') as f:
    f.write(json.dumps(data))


# # TOTAL CASES  MODEL

# In[12]:


y = df.total_cases_per_million
y.head()


# In[13]:


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 9)


# In[14]:


lr = LinearRegression()
```

```
lr.fit(X_train, y_train)
lr.score(X_test, y_test)


# In[15]:


with open('total_cases_model.pickle', 'wb') as f:
    pickle.dump(lr, f)


# # TOTAL DEATHS MODEL

# In[16]:


y = df.total_deaths_per_million
y.head()


# In[17]:


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
random_state = 9)


# In[18]:


lr= LinearRegression()
lr.fit(X_train, y_train)
lr.score(X_test,  y_test)


# In[19]:


with open('total_deaths_model.pickle', 'wb') as f:
    pickle.dump(lr, f)
```

Server

We have made a REST API service using the Flask framework. REST is an acronym for REpresentational State Transfer. It is an architectural style for distributed hypermedia systems and was first presented by Roy Fielding in 2000 in his famous dissertation.

```
from flask import Flask, request, jsonify
from flask_cors import CORS
import util

app = Flask(__name__)
CORS(app)

@app.route('/get_locations')
def get_locations():
    response = jsonify({
        'locations': util.get_locations()
    })

    response.headers.add('Access-Control-Allow-Origin', '*')
    return response

@app.route('/predict', methods=['POST'])
def predict():
    location = request.form['location']
    year = request.form['year']
    month = request.form['month']
    day = request.form['day']

    response = jsonify({
        'prediction': util.predict(location, year, month, day)
    })

    response.headers.add('Access-Control-Allow-Origin', '*')
    return response

if __name__ == "__main__":
    print('server starting')
    util.load_data_and_model()
    app.run()
```

```
import json
import pickle
import numpy as np

__locations = None
__columns = None
__total_cases_model = None
__total_deaths_model = None

def predict(location, year, month, day):
    loc_index = __columns.index(location)

    x = np.zeros(len(__columns))
    x[0] = year
    x[1] = month
    x[2] = day

    if loc_index >= 0:
        x[loc_index] = 1

    return {
        'total_cases': int(__total_cases_model.predict([x])[0]),
        'total_deaths': int(__total_deaths_model.predict([x])[0]),
    }

def get_locations():
    return __locations

def load_data_and_model():
    global __locations
    global __columns
    global __total_cases_model
    global __total_deaths_model

    with open('C:/Users/yash/Desktop/Mini Project/data.json',) as f:
        __columns = json.load(f)['columns']
        __locations = __columns[3:]

    with open('C:/Users/yash/Desktop/Mini Project/total_cases_model.pickle',
'rb') as f:
```

```
            __total_cases_model = pickle.load(f)

    with open('C:/Users/yash/Desktop/Mini Project/total_deaths_model.pickle',
'rb') as f:
            __total_deaths_model = pickle.load(f)

if __name__== "_main_":
    load_data_and_model()
```

CHAPTER 5

RESULTS/ OUTPUTS AND DISCUSSIONS

The images below are a series of all the web pages.

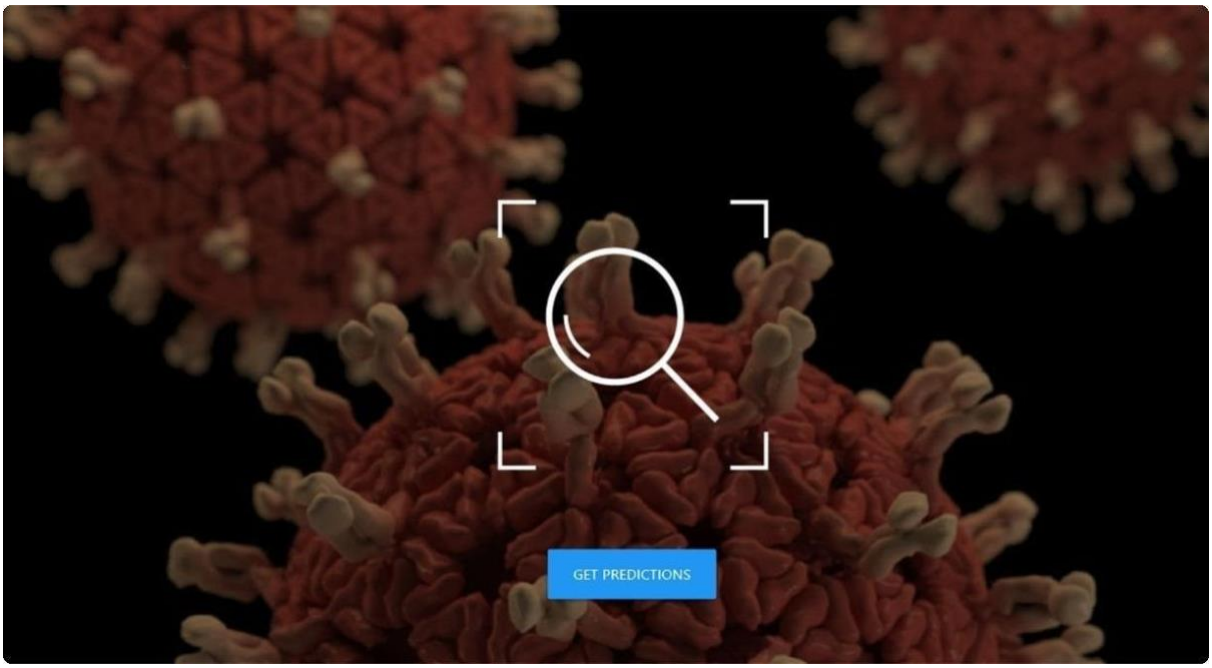


FIG 5.1: HOMEPAGE

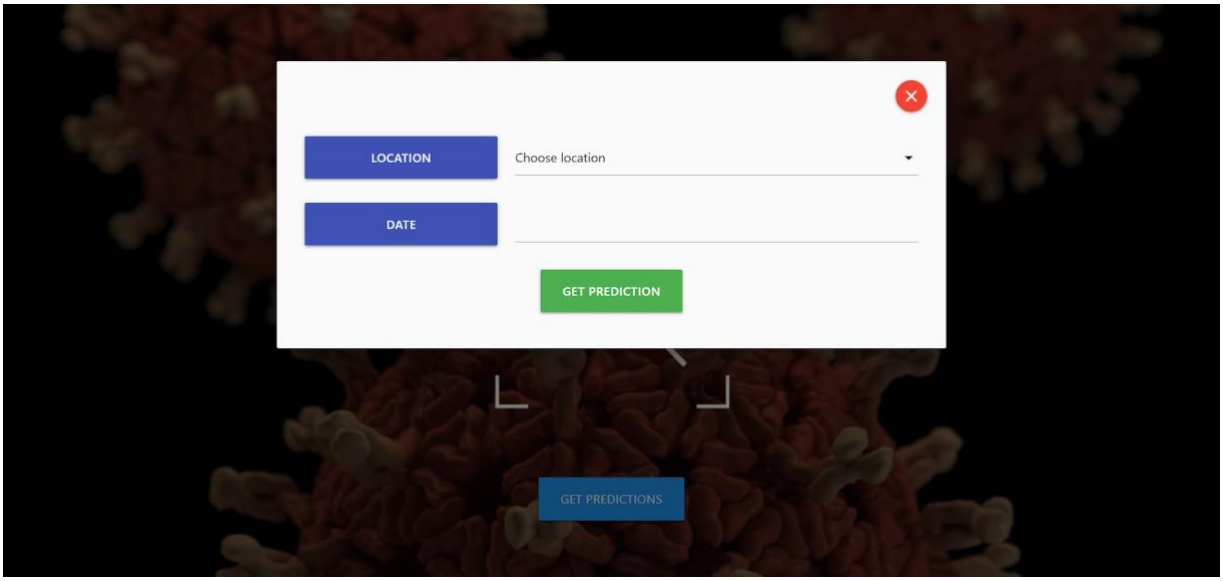


FIG 5.2: FORM TO GET THE ENTER THE LOCATION AND DATE

Locations selection

When the page renders it gets all the available locations from the server and displays them in a dropdown list.

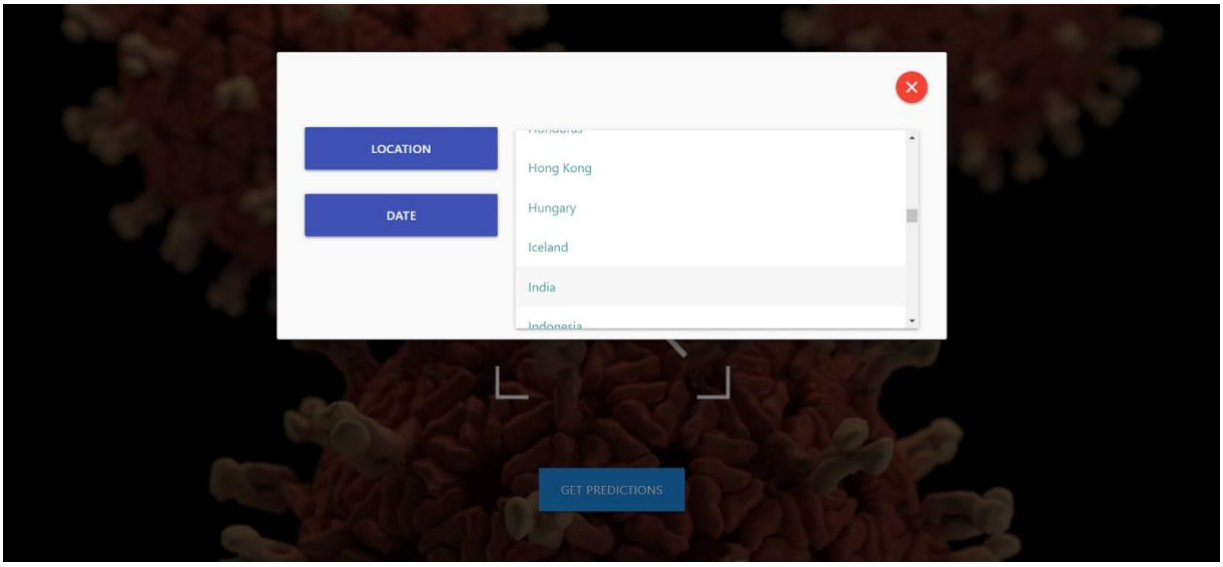


FIG 5.3.1: SELECTING LOCATION FROM THE DROPDOWN

Date selection

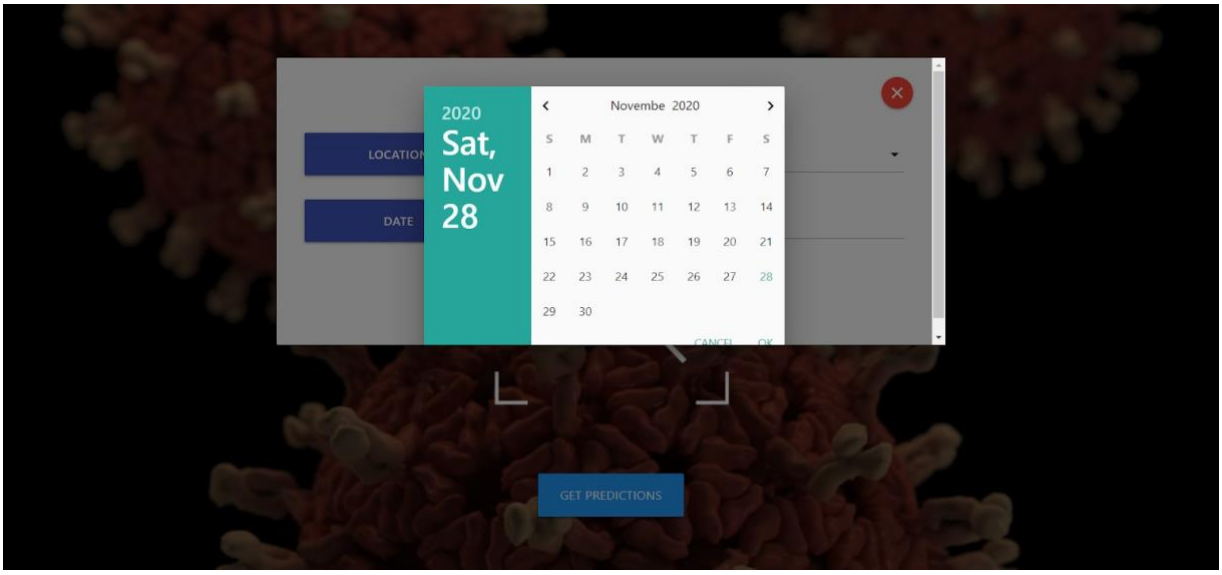
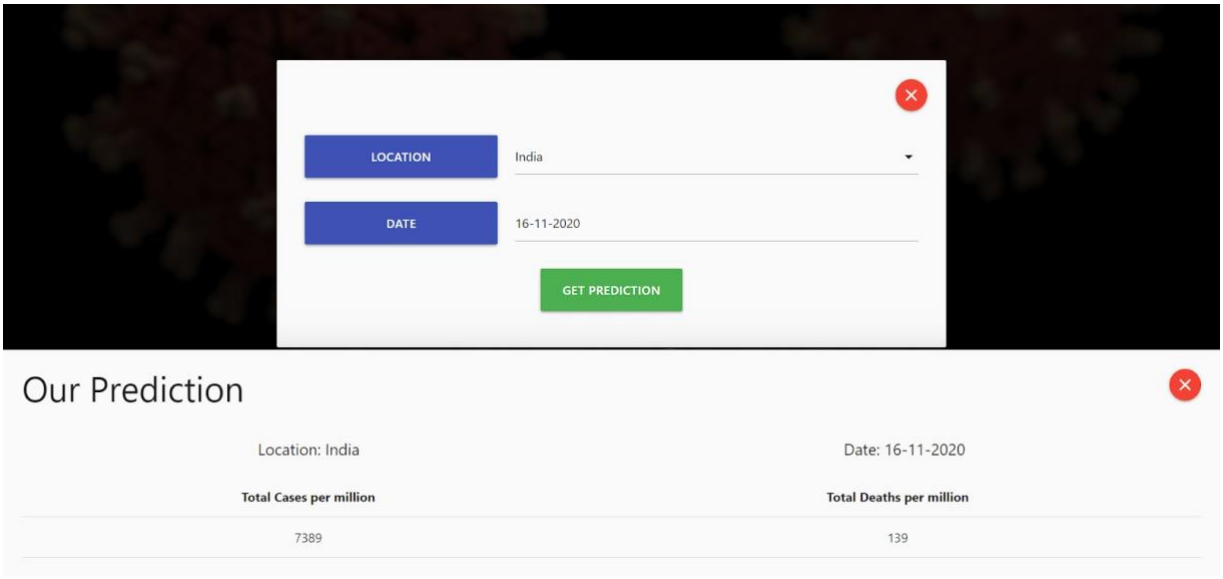


FIG 5.3.2: SELECTING DATE FROM DATEPICKER



Our Prediction

Location: India

Date: 16-11-2020

Total Cases per million	Total Deaths per million
7389	139

FIG 5.4: PREDICTED RESULT

CHAPTER 6

CONCLUSION AND FUTURE WORK

Conclusion

The main objective of this project is to create a COVID-19 prediction model which keeps a track of the current cases and makes a prediction about the future.

Limitations of the project

There are further areas where there is a possibility of improving the project like adding the Worst affected area feature so that the concerned authority can take appropriate options.

At this stage, the author is aware of many requirements of the project that are yet to be fulfilled. The ideas discussed below shall be implemented in the near future, and the project shall be kept alive with regular updates and modifications, that change and improve both the front-end and the back-end.

With better data sets the quality of the project can be further enhanced.

6.2. Future Scope

A lot of Organizations worldwide are coming up with various studies about the video which include:

Range of incubation periods for the disease in humans (and how this varies across age and health status) and how long individuals are contagious, even after recovery.

Prevalence of asymptomatic shedding and transmission.

Seasonality of transmission.

Physical science of the coronavirus (e.g., charge distribution, adhesion to hydrophilic/phobic surfaces, environmental survival to inform decontamination efforts for affected areas and provide information about viral shedding).

These studies will certainly bring out new datasets which will enhance the Output of our project.

CHAPTER 7
REFERENCES

1. https://flask-doc.readthedocs.io/en/latest/

2. https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f#:~:text=The%20purpose%20of%20holdout%20evaluation,used%20to%20build%20predictive%20models.

3. https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

4. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

5. https://ml-cheatsheet.readthedocs.io/en/latest/linear_regression.html#:~:text=Linear%20Regression%20is%20a%20supervised,Simple%20regression

6. https://www.geeksforgeeks.org/machine-learning-outlier/

7. https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/

8. https://api.jquery.com/

9. https://materializecss.com/

10. Datasets: https://ourworldindata.org/coronavirus-data