

A Project Report on

# **Surakhsha Kavach:ML based Cross Platform Application for Covid-19 Vulnerability Detection**

Submitted in partial fulfillment of the requirements for the award  
of the degree of

**Bachelor of Engineering**

in

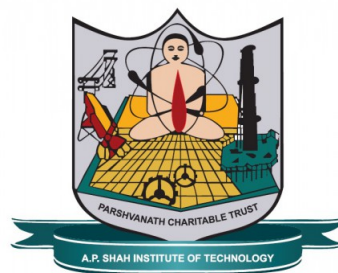
**Information Technology**

by

**Jasmine Kaur(18104010)  
Srushti Patil(18104061)  
Ruchi Raicha(18104068)**

Under the Guidance of

**Prof.Yaminee Patil  
Prof.Sonal Jain**



**Department of Information Technology  
NBA Accredited**

A.P. Shah Institute of Technology  
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615  
UNIVERSITY OF MUMBAI  
**Academic Year 2021-2022**

## Approval Sheet

This Project Report entitled “*Surakhsha Kavach:ML based Cross Platform Application for Covid-19 Vulnerability Detection*” Submitted by “*Jasmine Kaur*” (18104010), “*Srushti Patil*”(18104061), “*Ruchi Raicha*”(18104068)is approved for the partial fulfillment of the requirement for the award of the degree of *Bachelor of Engineering* in *Information Technology* from *University of Mumbai*.

(Name)  
Prof.Sonal Jain

(Name)  
Prof.Yaminee Patil

Prof. Kiran Deshpande  
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane  
Date:

## CERTIFICATE

This is to certify that the project entitled “*Surakhsha Kavach:ML based Cross Platform Application for Covid-19 Vulnerability Detection*” submitted by “*Jasmine Kaur*” (18104010), “*Srushti Patil*” (18104061), “*Ruchi Raicha*” (18104068) for the partial fulfillment of the requirement for award of a degree *Bachelor of Engineering* in *Information Technology*, to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

Prof.Sonal Jain  
Co-Guide

Prof.Yaminee Patil  
Guide

Prof. Kiran Deshpande  
Head Department of Information Technology

Dr. Uttam D.Kolekar  
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane

Date:

## Acknowledgement

We have great pleasure in presenting the report on “**Surakhsha Kavach:ML based Cross Platform Application for Covid-19 Vulnerability Detection**” We take this opportunity to express our sincere thanks towards our guide **Prof.Yaminee Patil** and **Prof.Sonal Jain** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Student Name1:Jasmine Kaur**  
**Student ID1:18104010**

**Student Name2:Srushti Patil**  
**Student ID2:18104061**

**Student Name3:Ruchi Raicha**  
**Student ID3:18104068**

## Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Signature)

---

(Jasmine Kaur (18104010))  
(Srushti Patil (18104061))  
(Ruchi Raicha (18104068))

Date:

## **Abstract**

The sector is facing a major health crisis due to the rapidity of Coronavirus transmission (covid-19). According to international health organization rules, the strongest anti-covid-19 protective measure carries a mask in public places and Crowded places. It's incredibly difficult to look at people with their hands in their Places. These days, RT-PCR is only one way to detect the COVID-19 infection, it is limited by the lack of time-consuming, and the need for specialized labs. However these 2 approaches cannot continuously be used for patients screening because of the radiation doses high costs and also the low variety of accessible devices hence there's a requirement for a less costly and quicker diagnostic model to spot the positive and negative cases of covid-19.

So in our project, we predict the vulnerability of peoples according to their age and states. In this paper, we used the four prediction models using four different classifiers (i.e Logistic regression, Naive byes, Random forest, SVM) for detecting the vulnerability of peoples from their age and state. We create all of four models with all the classifiers. The results showed that the Random forest classifier is the most accurate classifier for predicting the vulnerability of COVID-19 cases based on the age and state. The results could help in the early diagnosis of COVID-19, specifically when the RT-PCR kits are not sufficient for testing the infection and assist countries, specifically the developing ones that suffer from the shortage of RT-PCR tests and specialized laboratories. So this system is very helpful in COVID19. In this paper, no diagnostic model has been proposed to identify the positive and negative cases of COVID-19 using several clinical features. Therefore, this research aims to predict the COVID-19 positive or negative cases based on age and state using machine learning classification algorithms.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
<b>3</b>	<b>Objectives</b>	<b>5</b>
<b>4</b>	<b>Project Design</b>	<b>6</b>
4.1	Use case Diagram . . . . .	6
4.2	Activity Diagram . . . . .	7
4.3	Existing System Architecture . . . . .	8
4.4	Proposed System Architecture . . . . .	9
<b>5</b>	<b>Implementation</b>	<b>10</b>
5.1	English.json . . . . .	10
5.2	Home.dart . . . . .	11
5.3	Login.dart . . . . .	13
5.4	Register.dart . . . . .	15
5.5	Predict.dart . . . . .	17
5.6	Main.Dart . . . . .	20
<b>6</b>	<b>Testing</b>	<b>22</b>
6.1	Functional Testing . . . . .	22
6.1.1	Unit Testing . . . . .	22
6.1.2	Integration Testing . . . . .	22
6.2	Non-Functional Testing . . . . .	23
6.2.1	Compatibility Testing . . . . .	23
<b>7</b>	<b>Result</b>	<b>24</b>
7.1	Login Screen . . . . .	24
7.2	Home Screen . . . . .	25
7.3	Prediction Screen . . . . .	26
7.4	Localization Screen . . . . .	26
<b>8</b>	<b>Conclusion and Future Scope</b>	<b>28</b>
8.1	Conclusion . . . . .	28
8.2	Future Scope . . . . .	28
	<b>Bibliography</b>	<b>29</b>

<b>Appendices</b>	<b>30</b>
Appendix-A . . . . .	30
Appendix-B . . . . .	30
<b>Publication</b>	<b>31</b>



# List of Figures

2.1	Literature Review . . . . .	3
2.2	Literature Review . . . . .	4
2.3	Literature Review . . . . .	4
4.1	Use case diagram . . . . .	6
4.2	Activity diagram . . . . .	7
4.3	Existing System Architecture . . . . .	8
4.4	Proposed System Architecture . . . . .	9
7.1	Login and Register Screen UI . . . . .	24
7.2	Home Screen UI . . . . .	25
7.3	Prediction Screen UI . . . . .	26
7.4	Hindi,English,Marathi Language Localization Screen UI . . . . .	27

# List of Abbreviations

SVM: Support Vector Machine  
WHO: World Health Organisation  
SARSCOV-2: Severe Acute Respiratory Syndrome Coronavirus-2

# Chapter 1

## Introduction

On December 31, 2019, a novel pathogenic Coronavirus (2019- nCoV) epidemic was first discovered in Wuhan, Hubei Province, South China. On March 11th, 2020, the World Health Organization declared it a Pandemic. COVID-19 is the official name for the Coronavirus illness. COVID-19 is an infectious disease caused by the Coronavirus 2 (SARSCoV2) that causes severe acute respiratory syndrome. World Health Organization (WHO) declared the outbreak a Public Health Emergency of International Concern. As this covid is becoming a concern for all of us and its spreading very fast. Its necessary for us to take precautions as early as possible. So Suraksha kavach app is cross platform application where user can get the percentage attack of covid by entering his age and state. After he enters his age and state he gets the predicted percentage attack of covid. This prediction is done using various Machine Learning algorithms like Logistic Regression, Random Forest, Naïve Bayes and Support Vector Machine (SVM). According to the age and state the data is first fitted into training and testing, after getting probability it then classifies into 3 classes:- 0 - very less vulnerable 1- vulnerable 2- more vulnerable for example: If a user enters age as 60 and state as Maharashtra so his chances of getting covid is more as his age is more than 50 and also the covid cases are more so in output there will be message like "You are more vulnerable". This will help people especially the senior citizens to take precautions at a very early stage and protect themselves from getting covid and severe diseases

# Chapter 2

## Literature Review

1. In the first literature the author has discussed the regression analysis of Covid19 using various machine learning algorithms like polynomial regression, Linear Regression etc [1].
2. In the Second Literature the author has discussed the approach to predict corona virus using various machine Learning algorithms like Support Vector Machine(SVM),K-NN Algorithm,Decision Tree, Random Forest.
3. In the Third Literature the author has discussed regarding detection of covid-19 from medical images/Symptoms of Patients using machine learning.This has been carried out using deep learning algorithms.

Title	Authors	Advantages	Disadvantages	Result
<b>Predicting the COVID-19 infection with fourteen clinical features using machine learning classification algorithms</b>	<b>Ibrahim Arpaci &amp; Shigao Huang &amp; Mostafa Al-Emran &amp; Mohammed N. Al-Kabi &amp; Minfei Peng</b>	<b>The Precision, F-Measure, Recall, and Receiver Operating Characteristic (ROC) area metrics were used to evaluate the performance of the classifiers.</b>	<b>The problem of how to differentiate between positive and negative cases of COVID-19 is still a challenge that needs to be solved in order to curb the pandemic.</b>	<b>In line with these arguments and drawing on the bibliometric analysis results, no diagnostic model has been proposed to identify the positive and negative cases of COVID-19 using several clinical features.</b>

Figure 2.1: Literature Review

<b>Vulnerability Prediction From Source Code Using Machine Learning</b>	<b>MEHMET AKIF ERSOY, ELIF USTUNDAG SOYKAN</b>	<b>ML model for different vulnerability categories, which is an advantage in terms of training time, processing power, and memory requirements.</b>	<b>Another difficulty in vulnerability prediction is the class imbalance problem, arising from the fact that the number of vulnerable code samples is far less than the number of healthy code samples.</b>	<b>The true positive and false positive ratio for different binary AST depths that are used as a threshold to cut complete binary ASTs.</b>
---	--	---	---	---

Figure 2.2: Literature Review

<b>Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study</b>	<b><u>Davide Brinati</u>, <u>Andrea Campagner</u>, <u>Davide Ferrari</u>, <u>Massimo Locatelli</u>, <u>Giuseppe Banfi</u>, <u>Federico Cabitza</u></b>	<b>Algorithm confidence is 95%</b>	<b>It takes time to display output and also this project is not system compatible</b>	<b>Model can be used to discriminate among potential COVID-19 infectious patients with sufficient reliability, and similar sensitivity to the current Gold Standard.</b>
--	--	------------------------------------	---	--

Figure 2.3: Literature Review

# Chapter 3

## Objectives

- 1.To get the meaning of vulnerability to older persons themselves
- 2.To detect the vulnerability of a person based on age and state by using machine learning and various classification algorithms.
- 3.The idea of ML based app is to make people aware regarding their vulnerablities of covid attack which will help people to take precautions at an early stage to avoid further serious consequences and protect themselves and their family. Overall this app will be easily available on any platform which will help people to easily access and take the benefit of it. We have arrived at a conclusion that ML based Suraksha Kavach App is a much viable solution for the people to take precautionary measures at an early stage.
- 4.To build user Friendly Mobile Application

# Chapter 4

## Project Design

### 4.1 Use case Diagram

In this Project there are two models User and System. User can do registration, Login and view the output. System can do data preparation, data preprocessing, feature scaling and predict output

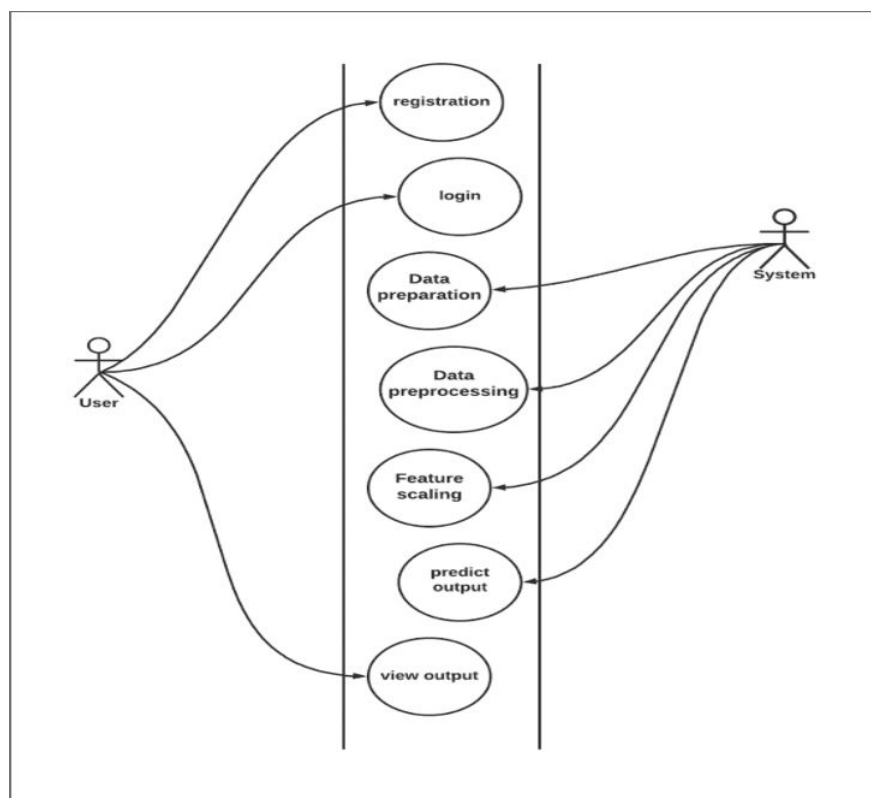


Figure 4.1: Use case diagram



## 4.2 Activity Diagram

So in this diagram user will first be able to see the login page if the user doesn't have an account then the user will have to first register. After the successful registration the user will be redirected to login page. After login the user will be able to see the home page. In home page there will be the button of check my vulnerability. Once the user clicks on this button the user will be redirected to prediction page.

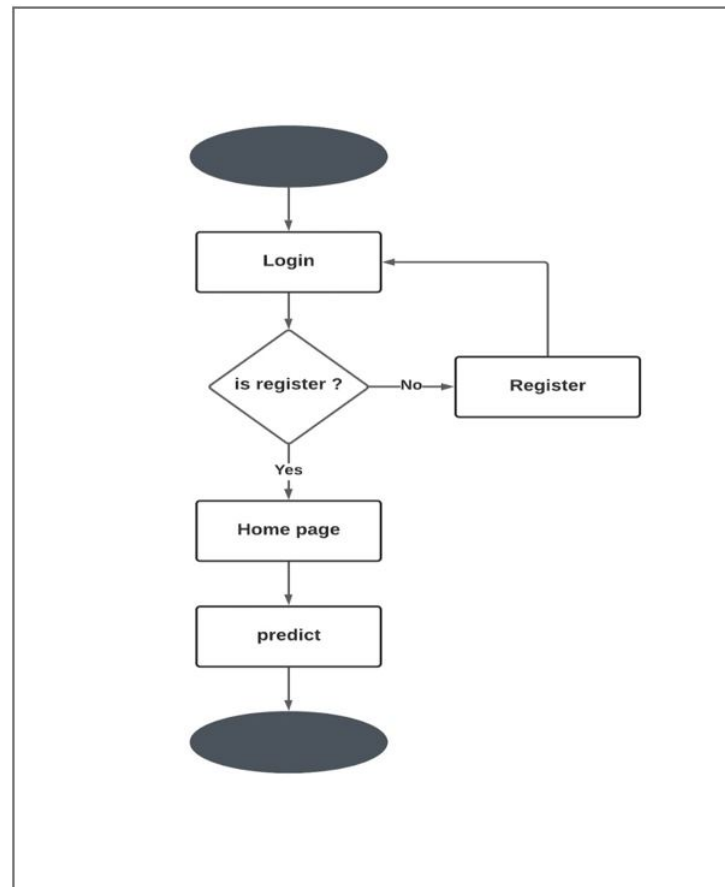


Figure 4.2: Activity diagram

### 4.3 Existing System Architecture

There are many Softwares available which gives information regarding covid cases and vaccination status. People may know that how many cases are around them with the help of tools. They may get the alert notification if there are too many covid cases around them So that a person can take a precautions. However there is no information regarding the percentage attack that one may have of covid which leads people to serious consequences. The Existing System includes-

- 1) The Registration
- 2) Login
- 3) User can see covid cases statewise
- 4) User can also see the information regarding vaccination
- 5) But there is no information regarding the Percentage of covid attack that one may have as a precautionary measure. This project aims to solve the limitations of the existing systems and further develop a new system.

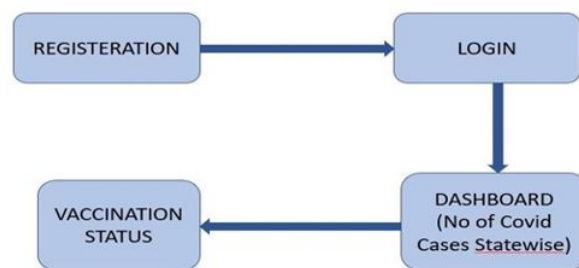


Figure 4.3: Existing System Architecture

## 4.4 Proposed System Architecture

Our Proposed System Architecture includes:

- 1) login/Register: It consist of username and password. this combination is all-inclusive authentication methods used because of convinence to others and low cost of deployment...Once user is login in...he/she can take experience of app.
- 2) Covid-19 Dashboard: Once the user is successfullly logged in....he/she can see the daily covid cases.
- 3) Check the vulnerability attack: The user can click on this button and can check his/her vulnerability attack of covid
- 4) Predict the attack: After clicking on the vulnerability attack button, enter his/her age and state and then he/she can get the vulnerability attack of covid he may have so that he can take precautionary measures at an early stage to prevent themselves from a serious diseases.The prediction is done based on ML algorithms.

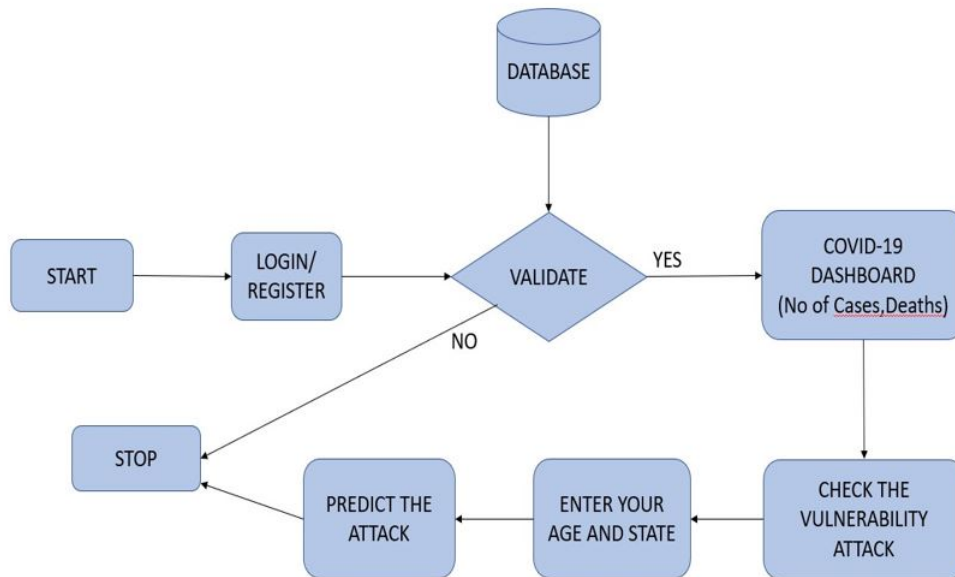


Figure 4.4: Proposed System Architecture

# Chapter 5

## Implementation

### 5.1 English.json

```
{
  "language": "English",
  "Login": "Login",
  "Login to your account": "Login to your\naccount",
  "Remember me": "Remember me",
  "OR": "OR",
  "Don't have an account? ": "Don't have an account? ",
  "Register": "Register",

  "Register new account": "Register new\naccount",
  "By creating an account, you agree to our": "By creating an account, you agree to o
  "Terms & Conditions": "Terms & Conditions",
  "Already have an account? ": "Already have an account? ",

  "Suraksha kavach": "Suraksha kavach",
  "Statistics": "Statistics",
  "My Country": "My Country",
  "Global": "Global",
  "Total": "Total",
  "Today": "Today",
  "Yesterday": "Yesterday",

  "Daily New Cases": "Daily New Cases",
  "Check my Vulnerability": "Check my Vulnerability",

  "Total Cases": "Total Cases",
  "1.81 M": "1.81 M",
  "Deaths": "Deaths",
```

```

    "105 K": "105 K",
    "Recovered": "Recovered",
    "391 K": "391 K",
    "Active": "Active",
    "1.31 M": "1.31 M",
    "Critical": "Critical",
    "N/A": "N/A",

    "Covid-19 Statistics": "Covid-19\nStatistics",
    "Predict": "Predict"
  }
}

```

## 5.2 Home.dart

```

class HomePage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        toolbarHeight: 60,
        title: LocaleText("Suraksha kavach",
          style: TextStyle(
            fontSize: 20.0,
          ),
        ),
        automaticallyImplyLeading: false,
        actions: <Widget>[
          IconButton(
            icon: Icon(
              Icons.logout,
              color: Colors.white,
              size: 30,
            ),
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => LoginPage()));
            },
          ),
        ],
      ),
    );
  }
}

```

```

    )
  ],
),
body: CustomScrollView(
  physics: ClampingScrollPhysics(),
  slivers: <Widget>[
    _buildHeader(),
    _buildRegionTabBar(),
    _buildStatsTabBar(),
    SliverPadding(
      padding: const EdgeInsets.symmetric(horizontal: 10.0),
      sliver: SliverToBoxAdapter(
        child: StatsGrid(),
      ),
    ),
    SliverPadding(
      padding: const EdgeInsets.only(top: 10.0),
      sliver: SliverToBoxAdapter(
        child: CovidBarChart(covidCases: covidUSADailyNewCases),
      ),
    ),
    SliverPadding(
      padding: const EdgeInsets.only(left: 20.0, right: 20.0, bottom: 20.0),
      sliver: SliverToBoxAdapter(
        child: ElevatedButton(
          style: ElevatedButton.styleFrom(
            primary: primaryBlue,
            onPrimary: Colors.white,
            shadowColor: Colors.black,
            elevation: 3,
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(15.0)),
            minimumSize: Size(500, 60),
          ),
          onPressed: (){
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => PredictPage()));
          },
        child: LocaleText('Check my Vulnerability',
          style: TextStyle(
            fontSize: 18.0,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
    ),
  ],
),

```

## 5.3 Login.dart

```
class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {

  var my1 = TextEditingController();
  var my2 = TextEditingController();

  bool passwordVisible = false;
  void togglePassword() {
    setState(() {
      passwordVisible = !passwordVisible;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: SingleChildScrollView(
          padding: EdgeInsets.fromLTRB(24.0, 40.0, 24.0, 0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  LocaleText(
                    'Login to your account',
                    style: heading2.copyWith(color: textBlack),
                  ),
                  SizedBox(
                    height: 20,
                  ),
                  Image.asset(
                    'assets/images/accent.png',
                    width: 99,
                    height: 4,
                  ),
                ],
              ),
              SizedBox(
```

```

        height: 48,
      ),
      Form(
        child: Column(
          children: [
            Container(
              decoration: BoxDecoration(
                color: textWhiteGrey,
                borderRadius: BorderRadius.circular(14.0),
              ),
              child: TextFormField(
                decoration: InputDecoration(
                  hintText: 'Username',
                  hintStyle: heading6.copyWith(color: textGrey),
                  border: OutlineInputBorder(
                    borderSide: BorderSide.none,
                  ),
                ),
                controller: my1,
              ),
            ),
            SizedBox(
              height: 32,
            ),
            Container(
              decoration: BoxDecoration(
                color: textWhiteGrey,
                borderRadius: BorderRadius.circular(14.0),
              ),
              child: TextFormField(
                obscureText: !passwordVisible,
                decoration: InputDecoration(
                  hintText: 'Password',
                  hintStyle: heading6.copyWith(color: textGrey),
                  suffixIcon: IconButton(
                    color: textGrey,
                    splashRadius: 1,
                    icon: Icon(passwordVisible
                      ? Icons.visibility_outlined
                      : Icons.visibility_off_outlined),
                    onPressed: togglePassword,
                  ),
                  border: OutlineInputBorder(
                    borderSide: BorderSide.none,
                  ),
                ),
                controller: my2,
              ),
            ),
          ],
        ),
      ),
    ),
  ),
),

```



```

        ),
      ),
    ],
  ),
),
 SizedBox(
  height: 32,
),
 Row(
  mainAxisAlignment: MainAxisAlignment.start,
  children: [
    CustomCheckbox(),
    SizedBox(
      width: 12,
    ),
    LocaleText('Remember me', style: regular16pt),
  ],
),
 SizedBox(
  height: 32,
),
 ElevatedButton(
  style: ElevatedButton.styleFrom(
    primary: primaryBlue,
    onPrimary: Colors.white,
    shadowColor: Colors.black,
    elevation: 3,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(15.0)),
    minimumSize: Size(500, 60),
  ),

```

## 5.4 Register.dart

```

class RegisterPage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

```

```

class _HomePageState extends State<RegisterPage> {

```

```

  var my1 = TextEditingController();
  var my2 = TextEditingController();
  var my3 = TextEditingController();
  var my4 = TextEditingController();

```

```

  bool passwordVisible = false;

```

```

void togglePassword() {
  setState(() {
    passwordVisible = !passwordVisible;
  });
}

String final_response = ""; //will be displayed on the screen once we get the data

final _formkey = GlobalKey<FormState>();

//function to validate and save user form
Future<void> _savingData() async{
  final validation = _formkey.currentState.validate();
  if (!validation){
    return;
  }
  _formkey.currentState.save();
}

//function to add border and rounded edges to our form
OutlineInputBorder _inputformdeco(){
  return OutlineInputBorder(
    borderRadius: BorderRadius.circular(20.0),
    borderSide:
    BorderSide(width: 1.0, color: Colors.blue, style: BorderStyle.solid),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    body: SafeArea(
      child: SingleChildScrollView(
        padding: EdgeInsets.fromLTRB(24.0, 40.0, 24.0, 0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                LocaleText(
                  'Register new account',
                  style: heading2.copyWith(color: textBlack),
                ),
                SizedBox(
                  height: 20,

```

```

    ),
    Image.asset(
      'assets/images/accent.png',
      width: 99,
      height: 4,
    ),
  ],
),
 SizedBox(
  height: 48,
),
 Form(
  child: Column(
    children: [
      Container(
        decoration: BoxDecoration(
          color: textWhiteGrey,
          borderRadius: BorderRadius.circular(14.0),
        ),
        child: TextFormField(
          decoration: InputDecoration(
            hintText: 'Username',
            hintStyle: heading6.copyWith(color: textGrey),
            border: OutlineInputBorder(
              borderSide: BorderSide.none,
            ),
          ),
        ),
      ),
    ],
  ),
),

```

## 5.5 Predict.dart

```

class PredictPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

```

```

class _LoginPageState extends State<PredictPage> {

```

```

  String dropdownValue = 'Select state';

```

```

  List <String> spinnerItems = [
    'Select state',
    'Haryana',
    'Rajasthan',
    'Uttar Pradesh',
    'Delhi',
    'Ladakh',
  ],

```

```

    'Tamil Nadu',
    'Kerala',
    'Jammu and Kashmir',
    'Karnataka',
    'Maharashtra',
    'Andhra Pradesh',
    'Telengana',
    'Punjab',
    'Uttarakhand',
    'Odisha',
    'West Bengal',
    'Gujarat',
    'Chhattisgarh',
    'Bihar',
    'Puducherry',
    'Tripura',
    'Jharkhand',
    'Assam',
    'Himachal Pradesh',
    'Manipur',
    'Mizoram',
    'Dadra and Nagar Haveli and Daman and Diu',
    'Goa',
    'Sikkim',
    'Arunachal Pradesh'
  ] ;

```

```

var my1 = TextEditingController();

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      toolbarHeight: 60,
      title: LocaleText("Suraksha kavach",
        style: TextStyle(
          fontSize: 20.0,
        ),
      ),
    ),
    automaticallyImplyLeading: false,
    actions: <Widget>[
      IconButton(
        icon: Icon(
          Icons.logout,
          color: Colors.white,
          size: 30,
        ),
      ),
    ],
  );
}

```

```

        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => LoginPage()),
          ),
        ),
      ],
    ),
    backgroundColor: Colors.white,
    body: SafeArea(
      child: SingleChildScrollView(
        padding: EdgeInsets.fromLTRB(24.0, 40.0, 24.0, 0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                LocaleText(
                  'Covid-19 Statistics',
                  style: heading2.copyWith(color: textBlack),
                ),
                SizedBox(
                  height: 20,
                ),
                Image.asset(
                  'assets/images/accent.png',
                  width: 99,
                  height: 4,
                ),
              ],
            ),
            SizedBox(
              height: 50,
            ),
            Form(
              child: Column(
                children: [
                  Container(
                    decoration: BoxDecoration(
                      color: textWhiteGrey,
                      borderRadius: BorderRadius.circular(14.0),
                    ),
                    child: TextFormField(
                      keyboardType: TextInputType.number,

```

```

        decoration: InputDecoration(
          hintText: 'Enter age',
          hintStyle: heading6.copyWith(color: textGrey),
          border: OutlineInputBorder(
            borderSide: BorderSide.none,
          ),
        ),
      ),
      controller: my1,
    ),
  ),
  SizedBox(
    height: 32,
  ),
  Container(
    decoration: BoxDecoration(
      color: textWhiteGrey,
      borderRadius: BorderRadius.circular(14.0),
    ),
    child:
      Padding(
        padding: EdgeInsets.fromLTRB(20, 5, 20, 5),
        child: DropdownButton<String>(
          isExpanded: true,
          value: dropdownValue,
          icon: Icon(Icons.arrow_drop_down),
          iconSize: 24,
          elevation: 16,
          style: TextStyle(color: Colors.black, fontSize: 17),
          onChanged: (String data) {
            setState(() {
              dropdownValue = data;
            });
          },
        ),
      ),
    ),
  ),

```

## 5.6 Main.Dart

```

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  void initState() {
    super.initState();
    Timer(Duration(seconds: 3),
      ()=>Navigator.pushReplacement(context,
        MaterialPageRoute(builder:

```

```

        (context) => LoginPage()
    )
  );
}
@override
Widget build(BuildContext context) {
  return Container(
    color: Colors.white,
    child: FlutterLogo(size: MediaQuery.of(context).size.height)
  );
}

```

# Chapter 6

## Testing

### 6.1 Functional Testing

#### 6.1.1 Unit Testing

The initial level of testing is unit testing, which is frequently carried out by the developers themselves. It is the process of ensuring that specific components of a piece of software are functional and work as intended at the code level. In a test-driven environment, developers will often write and execute the tests before passing the software or feature to the testing team. Manual unit testing is an option. Debugging will be easier as a result of unit testing since flaws will be detected earlier in the testing process and will take less time to fix than if they were discovered later. Our application development process is ideally suited for unit testing. During that time, we began to code in units to develop various modules. Also, test each module separately, such as the login, register page, Home Page, Prediction Page. All these pages are tested and debugged before going further integrating and check whether we are getting the desired output from each module as for the objectives.

#### 6.1.2 Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests. As we have discussed unit testing the next step is integration testing. All the units which we have tested and debugged are now ready to integrate into a whole single module. The integration part is crucial as we need to know which unit must interact without error, calling them in a different class accessing the instance of that class all these can be cleared with help of the activity diagram which was represented in Project Design. So accordingly, modules are integrated and checked whether they behave as for the objectives.



## **6.2 Non-Functional Testing**

### **6.2.1 Compatibility Testing**

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software's functionality is consistently supported across any environment you expect your end-users to be using. The framework we are using to develop our application is Flutter. It is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. We make sure that our application is compatible with both IOS and Android operating systems. The features we developed are perfectly run in multiple operating systems without an error. For this reason, compatibility testing is best suited for our project.

# Chapter 7

## Result

### 7.1 Login Screen

We have implemented a minimalistic UI that if a user firstly interacts with an application, it would be more attractive and user friendly. When the user interacts with the application first time he needs to register and then log into the application

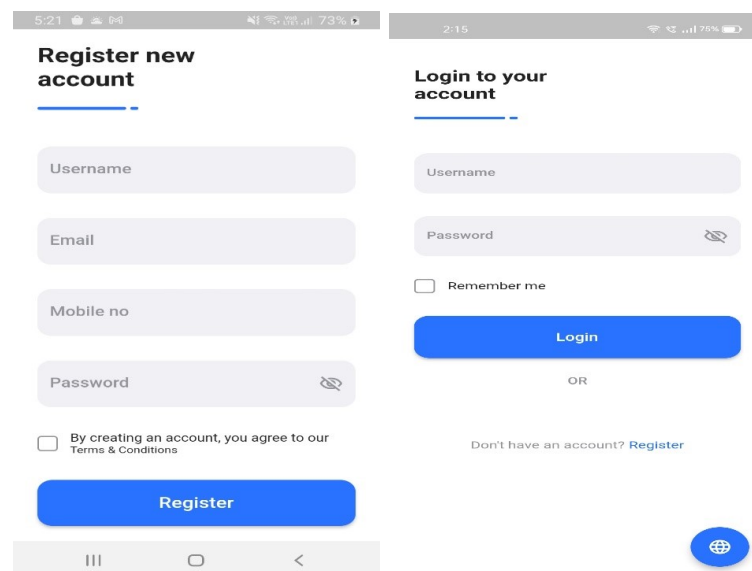


Figure 7.1: Login and Register Screen UI

## 7.2 Home Screen

After the user is successfully logged in the user will be able to see the Home Page which consists of covid cases in our country.(Static Page)

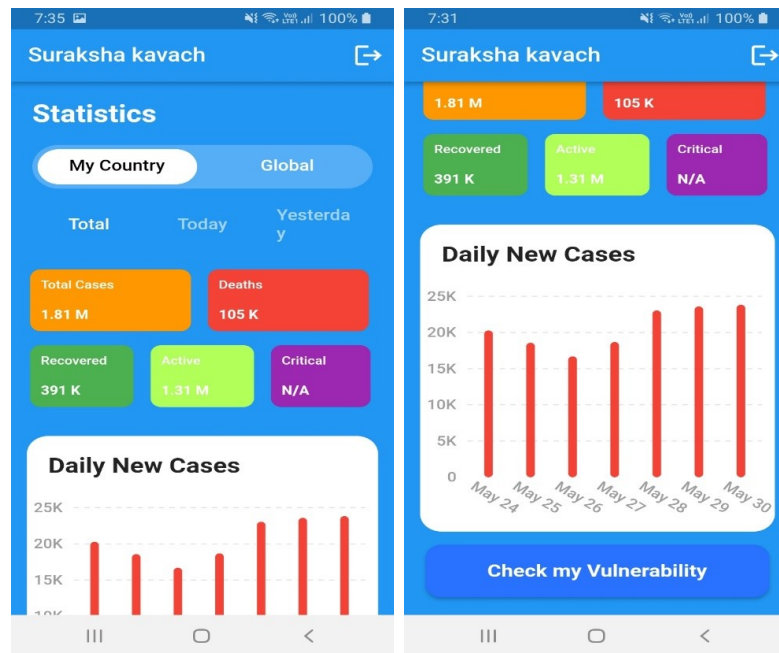


Figure 7.2: Home Screen UI

## 7.3 Prediction Screen

On Home Page there is a button of check my vulnerability attack once the user click on this button the user will need to enter his/her age and state and then click on predict button. After this the user will get to see the vulnerability attack of covid in him. So consider an example if a user enter his age as 60 and state as Maharashtra so the vulnerability attack of that person will be more because the cases in Maharashtra are more as compare to other states and also user age belongs to senior citizens. So once the user click on predict button he will get the output as “High Vulnerable” as shown below.

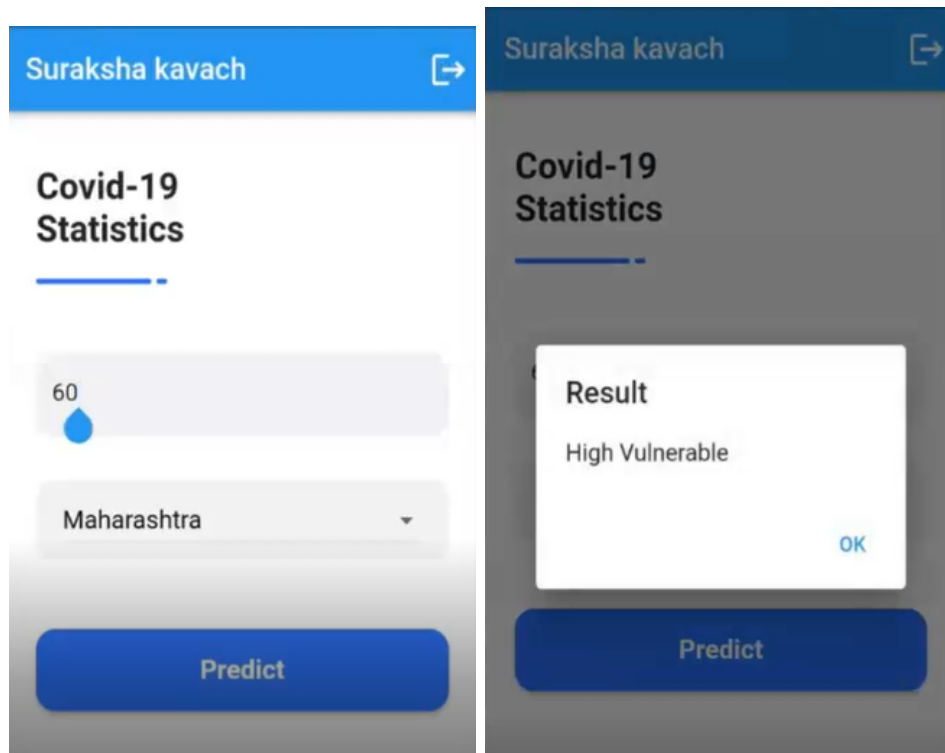


Figure 7.3: Prediction Screen UI

## 7.4 Localization Screen

There is also the feature of localization in our flutter based application where the user click on any of the languages such as English, Hindi or Marathi then the user will be able to see the UI in the respective language.



Figure 7.4: Hindi,English,Marathi Language Localization Screen UI

# Chapter 8

## Conclusion and Future Scope

### 8.1 Conclusion

In our project, the prediction of percentage attack is done using Machine Learning Algorithms. Random forest is suitable algo as its accuracy is more. We came up with this solution as people are getting infected by this covid at a very large number. So it's necessary to spread awareness and take precautions as soon as possible. This app will save the life of many people and help them to live happy and healthy life.

### 8.2 Future Scope

The idea about lockdown reminder is one of the features we can add so that people can know by the notifications on screen. The idea of booking vaccination slots can also be implemented.

# Bibliography

- [1] 1. Ibrahim Arpaci Shigao Huang Mostafa Al-Emran Mohammed N. Al-Kabi Minfei Peng “Predicting the COVID-19 infection with fourteen clinical features using machine learning classification algorithms ” Multimedia Tools and Applications (2021)
- [2] 2. Kolla Bhanu Prakash , S. Sagar Imambi , Mohammed Ismail , T Pavan Kumar , YVR Naga Pawan “Analysis, Prediction and Evaluation of COVID-19 Datasets using Machine Learning Algorithms” Kolla Bhanu Prakash et al., International Journal of Emerging Trends in Engineering Research, 8, May 2020
- [3] 3. Davide Brinati ,Andrea Campagner ,Davide Ferrari ,Massimo Locatelli ,Giuseppe Banfi ,Federico Cabitza “ Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study ” 23 April 2020
- [4] 4. Ioannis D. Apostolopoulos , Tzani A. Mpesiana “ Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks ” 25 March 2020

# Appendices

## Appendix-A: : Flutter Download and Installation

1. Download flutter dk from [ps://docs.flutter.dev/development/tools/sdk/releases?tab=windows](https://docs.flutter.dev/development/tools/sdk/releases?tab=windows)
2. Place flutter sdk in your desired directory; like /user/Documents.
3. Go to environment variable set path for flutter.
4. Download Android Studio from <https://developer.android.com/studio> 6. Double click .exe file for installation.
5. Double click .exe file for installation.

## Appendix-B: : Anaconda Download and Installation

1. Download Anaconda from <https://www.anaconda.com/products/distribution>



# Publication

Paper entitled **“Surakhsha Kavach : ML based Cross Platform Application for Covid-19 Vulnerability Detection”** is presented at **“ICTIS 2022 6th International Conference on ICT for Intelligent Systems”** by **“Jasmine Kaur Wadhwa”, “Ruchi Raicha”, “Srushti Patil”**.