VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report

on

Covid-19 Tracker App

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

by

Supriya Alavala

1RN18IS110

Under the Guidance of

Mrs. Shwetha G N

Assistant Professor

Department of ISE

ESTD:2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,

Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Covid-19 Tracker App*** has been successfully completed by **Supriya Alavala(1RN18IS110)** bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

____________________	____________________	____________________
Mrs. Shwetha G N	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. ____________________________	**1.** ______________________
2. ____________________________	**2.** ______________________

DECLARATION

I, **Supriya Alavala [USN: 1RN18IS110]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Covid-19 Tracker App*** has been carried out by me and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University*, *Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

SUPRIYA ALAVALA

(1RN18IS110)

ABSTRACT

The World Health Organization has declared the outbreak of the novel coronavirus, Covid-19 as pandemic across the world. With its alarming surge of affected cases throughout the world, lockdown, and awareness (social distancing, use of masks etc.) among people are found to be the only means for restricting the community transmission. In a densely populated country like India, it is very difficult to prevent the community transmission even during lockdown without social awareness and precautionary measures taken by the people.

COVID-19 has become biggest impediment for the survival of the human race, at present. Again, as mobile technology is now an important component of human life, hence it is possible to use the power of mobile technology against the treat of COVID-19. Every nation is now trying to deploy an interactive platform for creating public awareness and share the important information related to COVID-19. Keeping all of these in mind, an attempt to deploy an interactive cross-platform (web/mobile) application COVID-19 TRACKER for the ease of the users.

The application is featured with all the real-time attributes about the novel coronavirus disease and its measures and controls. To achieve all these functionalities, many tools and APIs are used in this application. The system purposely aims to maintain the digital protection of the society, create public awareness, and not create any agitation situation among the individuals of the society.

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mrs. Shwetha GN** Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R, Co-Founder & CEO of ENMAZ**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF ABBREVIATIONS

COVID-19	- Corona Virus Disease 19
WHO	- World Health Organization
PHEIC	- Public Health Emergency of International concern
REST API	- Representational State Transfer Application Programming Interface
UI	- User Interface
SOAP	- Simple Object Access Protocol
HTTP	- Hypertext Transfer Protocol
BLOC	- Business Logic Component
JSON	- JavaScript Object Notation
JS	- JavaScript
JSP	- Java Server Pages
ASP	- Active Server Pages
BSD	- Berkeley Software Distribution

1. INTRODUCTION

1.1 Background

In Wuhan city of China, due to some unknown causes a local outbreak of pneumonia was notice in December 2019. Originally, the disease got named serious acute respiratory syndrome coronavirus 2 (SARS-COV-2). Corona Virus Disease 19 (COVID-19) caused by the novel coronavirus is confirmed officially later by the World Health Organization (WHO) and they also declare COVID-19 as the Public Health Emergency of International Concern (PHEIC). This ongoing outbreak of pneumonia spread rapidly all over the globe, causing more than 1,83,820 deaths soon got identified as the cause of the novel coronavirus. As of April 25, 2020, more than 22,398 confirmed COVID-19 cases found in INDIA. Social distancing was found to be a promising way to saturate the growth of the disease. Thus, most of the nations declared lockdown of the entire state to set a bar on the growing numbers of the disease. But it was found 42% of the society was not aware of the pandemic yet, and the lockdown caused a huge economic break.

Keeping all of these complex features in mind we have used all the promising technologies to build the mobile application as well as a web application. The application is capable to provide real-time worldwide cased. The application also provides information on different preventive measures to be taken by users and symptoms of COVID-19.

Nowadays, wireless communication in terms of mobile technology is an essential part of our life. This technology is now the guiding force for human life to perform any activities. Again, in this pandemic situation, fake news prevention and distribute genuine information, i.e., information regarding public awareness, test centers, geo-tracking, etc. to everyone in every time with minimum expense, it is perhaps the biggest challenge to any administration all over the world. Mobile technology is the only one stop solution for everything above mentioned problem.

1.2 Existing Systems

In this pandemic condition, a couple of mobile applications are introduced by government institutions and others, which are providing valuable information to the people to aware of the pandemic situation, like the number of confirmed cases, number of recovered, number of deaths as of now, etc. APOLLO, one of the biggest medical giants in India, developed a risk assessment scanner for COVID-19 outbreak in INDIA and named it “covid. apollo247”.

This risk assessment scanner comes with a tag line “Stay calm amidst the current paranoia surrounding COVID-19” and is successful in reaching a major part of the society during this outbreak. The solution could efficiently predict the chances of risk (Low, Mid, High) of having positive coronavirus tress depending on multiple users given parameters using their ML algorithms. A team of twenty plus members (teachers and students) from Mahindra Ecole Centrale developed a dashboard for smooth user experience and quality visualization of real-time data. The website shows the real-time data of the infected, cured, and deaths in INDIA, though some inconsistency is also observed in their application. Several graphs and charts are used in the website to visualize the data also at the district level, but more accuracy is expected concerning their results.

The Government of India took the initiative and developed a mobile application named AarogyaSetu to connect essential health services with the people of INDIA. Basically, it is a mobile application; still, the use of web views was found in the statistics display dashboard. They have also come up with a geo-tracking facility that helps the system to track the coronavirus affected people and form a cluster. MICROSOFT BING has taken the novel attempt to implement a software solution for the COVID-19 pandemic across the world for data transparency and increasing traceability. Bing map comes with an interactive platform that focused on the world map. When hovering over the different countries, it used to render the COVID-19 statistics (Confirmed cases, Active cases, Deaths, Recovered) of the particular nation.

1.3 Proposed System

An application with a lot of information may be confusing for the user with less knowledge on how to use mobile/web. We propose a simple Covid-19 Tracker which provides most important functionalities. The application uses API to fetch the real-time number of active cases, number of recovered and deaths. It provides means to find any country of his choice and allow the user to set a default country for frequent updates.

The application provides users information on top COVID-19 symptoms and precautions to be taken to prevent from getting affected. It also tries to burst some myths related to COVID-19 and provides important information about the virus.

2. LITERATURE REVIEW

In this paper [1], the problem of infodemia (an overload of information about a problem, usually false and unverified) created by declaring a COVID-19 pandemic is considered. A list of Web services has been formed that provide reliable pandemic data from relevant sources, and as such, in the fight against the COVID-19 infodemia can be used. The freemium Web API service COVID-19 data in detail with a set of GET methods by which users can request reliable data is described. In addition to global pandemic data, data on individual countries and regions from this service may be requested. The latency of the Web API service COVID-19 data was determined using the getLatestTotals method for reading global data. Based on the obtained results, a way of using this Web service in the fight against the COVID-19 infodemia was proposed.

In today's world, reliability on the web services are degrading every day, hence this paper tried to find a solution this problem hence the web api will become more reliable in giving us the day-to-day covid-19 information which will lead to a reliable application.

In this paper [2], presents development of smart e-health system for Covid-19 pandemic. It is a smart Telemedicine system where patient can consult with doctors staying at home. Real-time online Doctor-Patient interaction and prescription are the main features. Increasing popularity of online systems and to avoid time waste, distance people will easily take advantage of this it. The doctors are specialized and highly professional in their field. Website has also blog, shop site where doctor post different health issue to make awareness among the society and patient can buy their medicine as well.

Health is very important in one’s life. This paper speaks that covid-19 has really brought the online era, this covid-19 tracker application is one step towards that era where people use the online means to get consultation and similarity this app will help them keep in track with the current situations in various places of the world.

In this Paper [3], as we all know that during this tough time everyone is facing the summons because of this scary pandemic Covid19. More than 200 countries are affected due to this pandemic and the populace is uninformed when this drastic period will come to an end. How the rapid growth of this virus is going to be stopped, apart from vaccinations and Medicare everyone needs to be much more aware about the dangerous situation and must follow the protocols and guidelines imposed by government in order to be safe or free from the scary virus.

In this paper the work was to build an android platform to layout mindfulness about the pandemic and furthermore to help the world population by providing them the useful information regarding the coronavirus. An android app which will be helpful in displaying all the data of Covid19 (such as number of total cases worldwide, number of active cases, covid19 hospitals around the covid19 victims, medical facilities near them and many more things. Getting real time data for anything is very beneficial for all the users, as we are displaying the data for covid19 so without going anywhere they can easily get to know their current place status for corona i.e. total number of covid19 cases in their country or city. This way of collecting data will be helpful as the users will not come in direct contact with anyone and hence, they do not get affected.

As we all know that growing pandemic doesn't seem to be endemic any time soon. Hence this paper explains the need of a application to keep in track of the everyday situation in various places of the world, that where the covid-19 tracker application will play a vital role.

In this Paper [4], nowadays Internet has become the most prominent source of information for searching the data in health sector for medical purpose fulfilments. However, in this present time this task is like a herculean task by lack of implementation of the proper skill framework and complexities that lies in Computer science and Engineering era. This paper is to computerize the Management of Hospitals to upgrade the software system that is user friendly, cost effective, fast, and simple, and to store and register the patient details and the doctor details. The aim is to develop a processed hospital management system which will upgrade the potency of the service provided to patient's mistreatment and to ensure that the adequate information is available. The information may be retrieved simply as the data will be stored in the database. The information is fully protected for private use and completes the process in less time.

The population in India is increasing rapidly and loads and loads of experts have observed the requirement of information availability in the field of healthcare. This application will help us keep track of the enormous data and help the public in providing the awareness of the outbreak and in turn bring serious in of the pandemic.

In this paper[5], flutter is a popular UI framework for developing mobile applications by Google. It has caught traction in recent years. However, Flutter developers have to deal with a state management issue when developing their applications. In order to solve this problem, multiple architectures have been developed. This paper proposes a new Flutter architecture based on the Clean Architecture by Uncle Bob.

The Flutter Clean Architecture proposed in this paper is packaged and released through a Flutter package. The architecture is tested by developing a full application from scratch using the package and documenting the process. The Flutter Clean Architecture provides a solution to the state management problem as well as a potential overall choice for Flutter mobile application architecture.

In today's world, there are many new technologies arising. Flutter is one of those upcoming technology. In paper tries to resolve the many issues faced while using flutter making flutter on of the most reliable and user-friendly application to use.

3. ANALYSIS

3.1 Introduction

In general, developing a mobile application is a complex and challenging task. There are many frameworks available to develop a mobile application. Android provides a native framework based on Java language and iOS provides a native framework based on Objective-C / Swift language.

However, to develop an application supporting both the OSs, we need to code in two different languages using two different frameworks. To help overcome this complexity, there exists mobile frameworks supporting both OS. These frameworks range from simple HTML based hybrid mobile application framework (which uses HTML for User Interface and JavaScript for application logic) to complex language specific framework (which do the heavy lifting of converting code to native code). Irrespective of their simplicity or complexity, these frameworks always have many disadvantages, one of the main drawbacks being their slow performance.

In this scenario, Flutter – a simple and high-performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native framework.

3.1.1 Flutter

Flutter is a cross-platform software development framework that was presented by Google in 2015 and received its first release in May of 2017. Flutter nowadays has steadily grown and provided possibilities not only for iOS and Android mobile development but also for web and desktop applications as well.

Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML.

To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and new) and render the widget with only the necessary changes instead of re-rendering the whole widget.

Flutter framework offers the following features to developers −

- Modern and reactive framework.
- Uses Dart programming language and it is very easy to learn.

- Fast development.
- Beautiful and fluid user interfaces.
- Huge widget catalogue.
- Runs same UI for multiple platforms.
- High performance application.

Figure 3.1
Introduction

3.1.2 Flutter REST API

Representational State Transfer (REST) is an architectural style that defines a set of constraints to be used for creating web services. REST API is a way of accessing web services in a simple and flexible way without having any processing.

REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST uses less bandwidth, simple and flexible making it more suitable for internet usage. It’s used to fetch or give some information from a web service. All communication done via REST API uses only HTTP request.

Today, most of the apps use remote data using APIs. So, this section will be the important part for those developers who want to make their carrier in Flutter.

Flutter provides http package to use http resources. The http package uses await and async features and provides many high-level methods such as read, get, post, put, head, and delete methods for sending and receiving data from remote locations. These methods simplify the development of REST-based mobile applications.

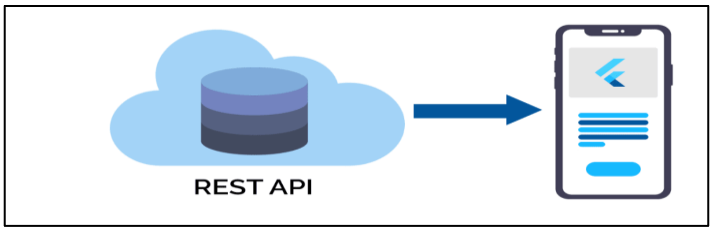


Figure 3.2
Rest Api

3.2 Software requirement specification

The best thing about using Flutter for creating cross-platform native mobile apps is the fact that you can build those on almost any OS.

Here are some System Requirements for Android Studio which is needed for running an Android simulator.

Windows:

- Microsoft® Windows® 7/8/10 (64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum,
- 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems**: Windows 7 SP1 or later (64-bit), x86-64 based.
- **Disk Space**: 1.64 GB (does not include disk space for IDE/tools).
- **Tools**: Flutter depends on these tools being available in your environment.
    - Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10)
    - Git for Windows 2.x, with the Use Git from the Windows Command Prompt option.

If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.

4. SYSTEM DESIGN

4.1 Flutter Architecture

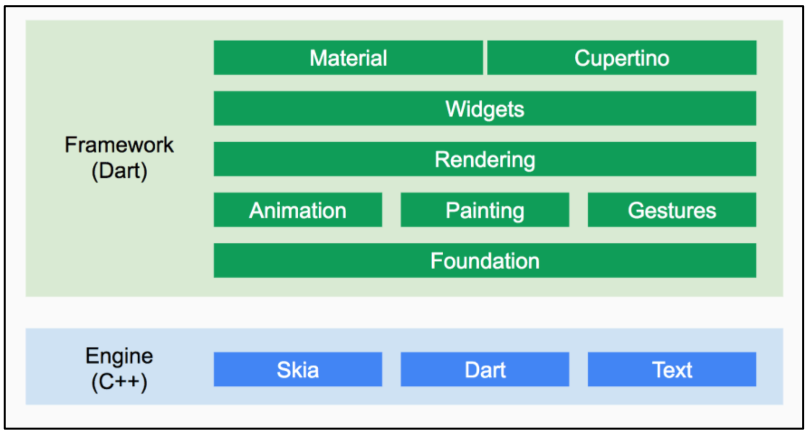


Figure 4.1
Flutter Architecture

Flutter is organized around layers. Each layer is built upon the previous.

From the diagram we can see the low-level part of Flutter is an Engine built in C++. It provides low-level rendering support using Google's Skia graphics library.

The high-level part of the diagram is the Framework written in Dart. It provides libraries to handle animation, gestures, rendering, widgets and more.

With all this layer the developer can do more with less code by using elements on the top or go down to customize some behavior of its app.

Everything is a widget

In Flutter, everything is a widget nested inside another widget. It comes with beautiful, customizable widgets and we can control the behavior of each widget and also styling becomes easy.

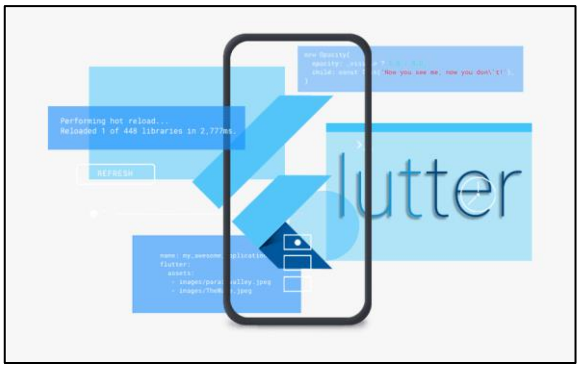


Figure 4.2
Everything is a widget

All the widget of a Flutter app forms a hierarchy where a widget is a composition of other widgets and each widget inherits properties from its parent.

4.1.1 Basic widgets

Flutter comes with a suite of powerful basic widgets, of which the following are commonly used:

Text:

The Text widget lets you create a run of styled text within your application.

Row, Column:

These flex widgets let you create flexible layouts in both the horizontal (Row) and vertical (Column) directions. The design of these objects is based on the web's flexbox layout model.

Stack:

Instead of being linearly oriented (either horizontally or vertically), a Stack widget lets you place widgets on top of each other in paint order.

Container:

The Container widget lets you create a rectangular visual element.

Appbar:

A Material Design app bar. An app bar consists of a toolbar and potentially other widgets, such as a TabBar and a FlexibleSpaceBar.

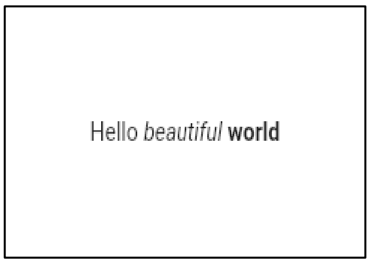


Figure 4.3
Text

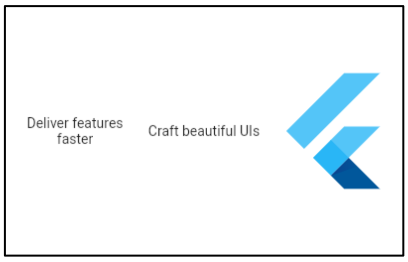


Figure 4.4
Row

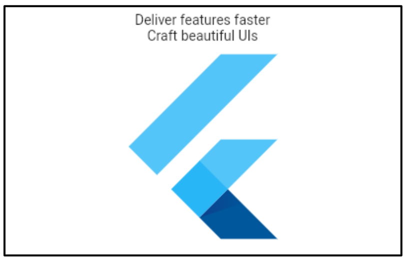


Figure 4.5
Column

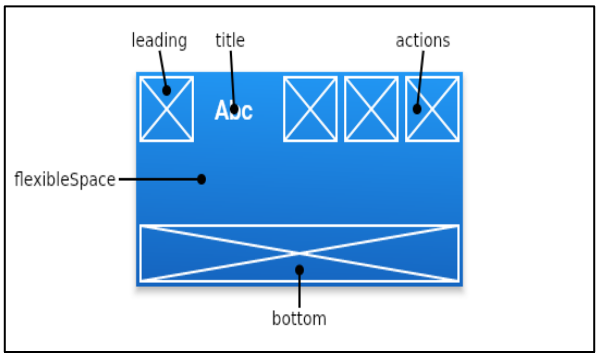


Figure 4.6
AppBar

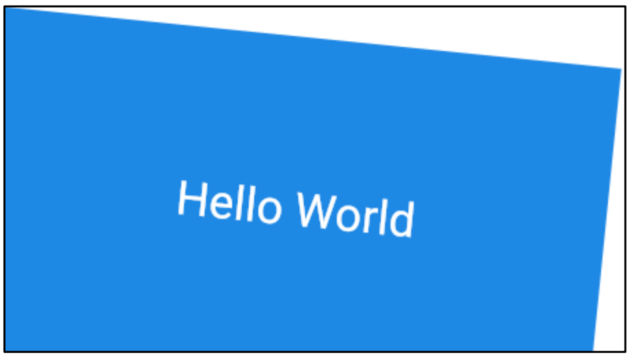


Figure 4.7
Container

4.1.2 What is Material design?

Material is an adaptable design system, backed by open-source code, that helps developers easily build high-quality, digital experiences. From design guidelines to developer components, Material can help developers build products faster. Material design guidelines provide best practices and user interface design.

1. App Structure and navigation

It consists of:

- Appbar
  A Material Design app bar. An app bar consists of a toolbar and potentially other widgets, such as a TabBar and a FlexibleSpaceBar.

- Drawer
  A Material Design panel that slides in horizontally from the edge of a Scaffold to show navigation links in an application.

- MaterialApp
  A convenience widget that wraps a number of widgets that are commonly required for applications implementing Material Design.

- Scaffold
  Implements the basic Material Design visual layout structure. This class provides APIs for showing drawers, snack bars, and bottom sheet.

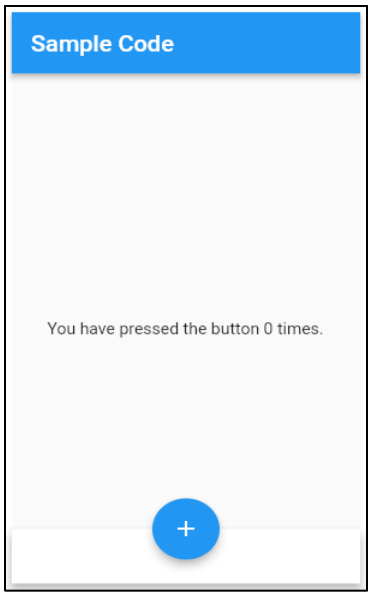


Figure 4.8
Scaffold

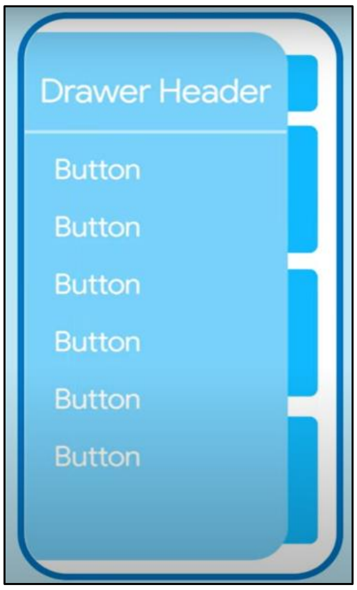


Figure 4.9
Drawer

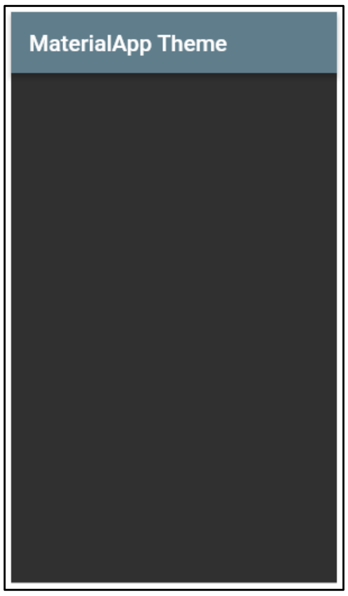


Figure 4.10
MaterialApp

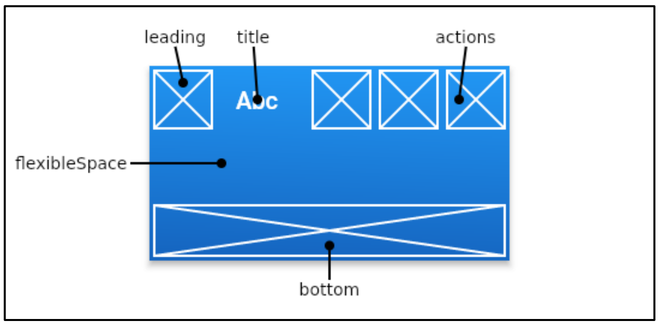


Figure 4.11
AppBar

2. **Buttons**

- <u>DropdownButton</u>

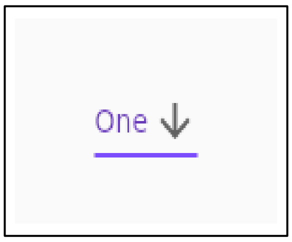
  Shows the currently selected item and an arrow that opens a menu for selecting another item.

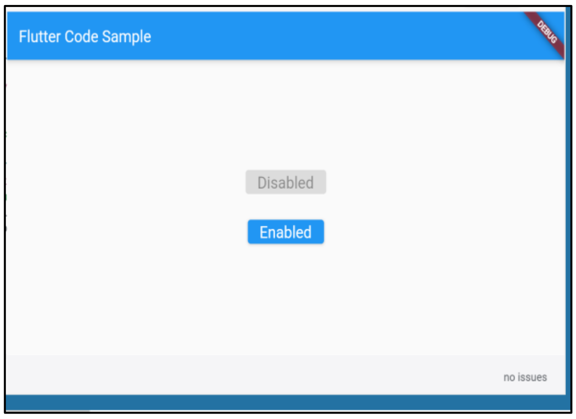
- <u>ElevatedButton</u>

  A Material Design elevated button. A filled button whose material elevates when pressed.

- <u>FloatingActionButton</u>

  A floating action button is a circular icon button that hovers over content to promote a primary action in the application.





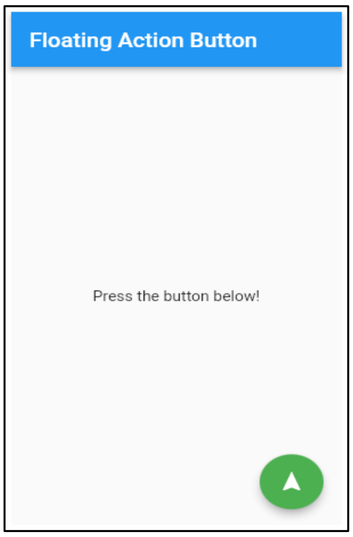


Figure 4.12
Dropdown Button

Figure 4.13
Elevated Button

Figure 4.14
Floating Action Button

3. **Input and selections**

- <u>Checkbox</u>

  Checkboxes allow the user to select multiple options from a set. The Checkbox widget implements this component.

- <u>Slider</u>

  Sliders let users select from a range of values by moving the slider thumb.

- Text Field

  Touching a text field places the cursor and displays the keyboard. The Text Field widget implements this component.

- Date & Time Pickers

  Date pickers use a dialog window to select a single date on mobile.

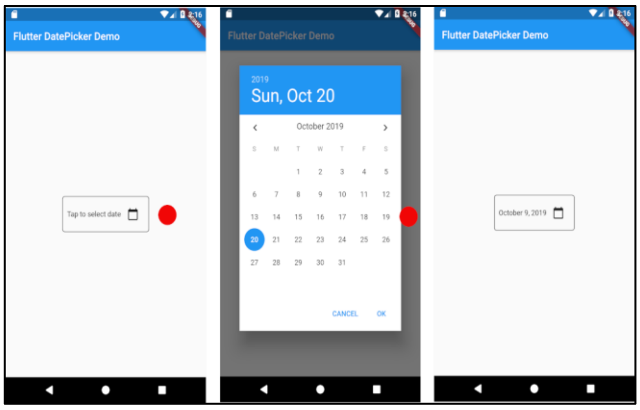


Figure 4.15
Date Picker

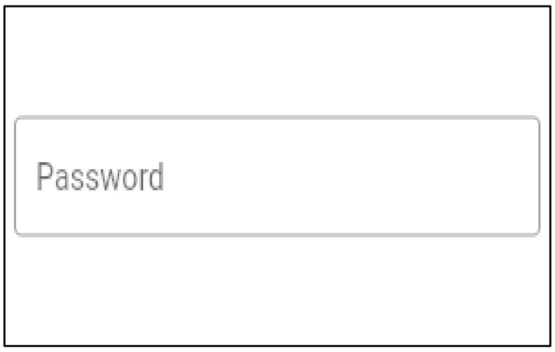


Figure 4.16
Text Field

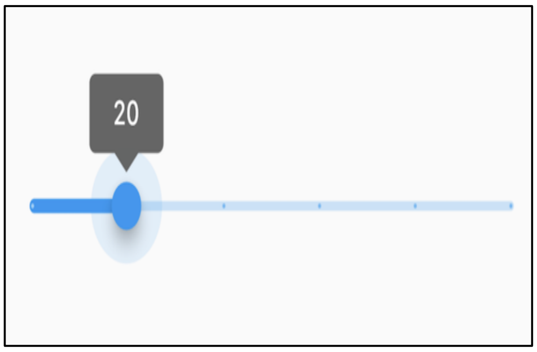


Figure 4.17
Slider

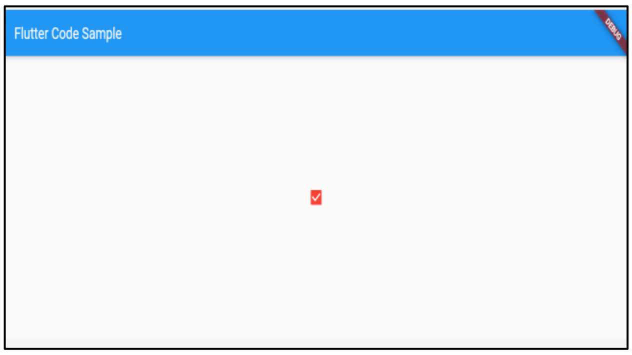


Figure 4.18
Checkbox

4. Layout

- Divider

  A one logical pixel thick horizontal line, with padding on either side.

- ListTile

  A single fixed-height row that typically contains some text as well as a leading or trailing icon.

- Stepper

  A Material Design stepper widget that displays progress through a sequence of steps.

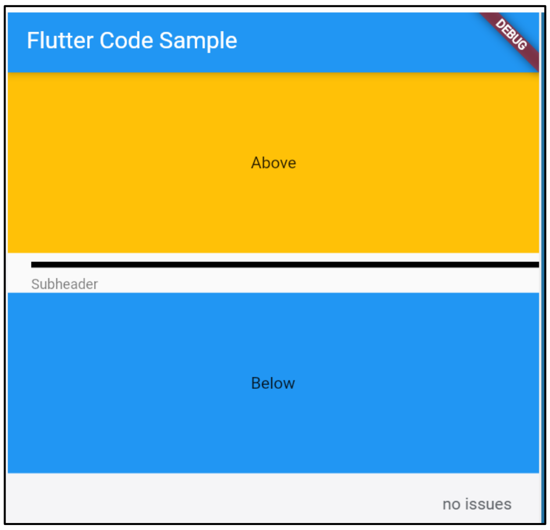


Figure 4.19
Divider

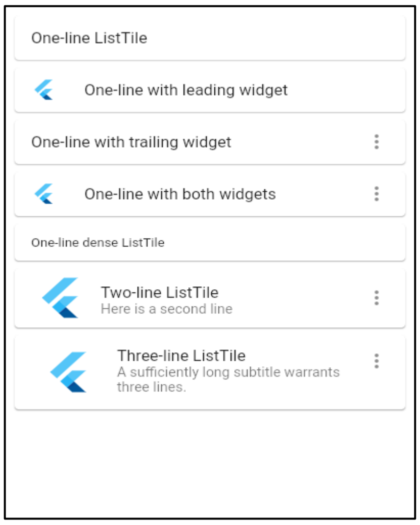

One-line ListTile
One-line with leading widget
One-line with trailing widget
One-line with both widgets
One-line dense ListTile
Two-line ListTile
Here is a second line
Three-line ListTile
A sufficiently long subtitle warrants three lines.

Figure 4.20
List Tile

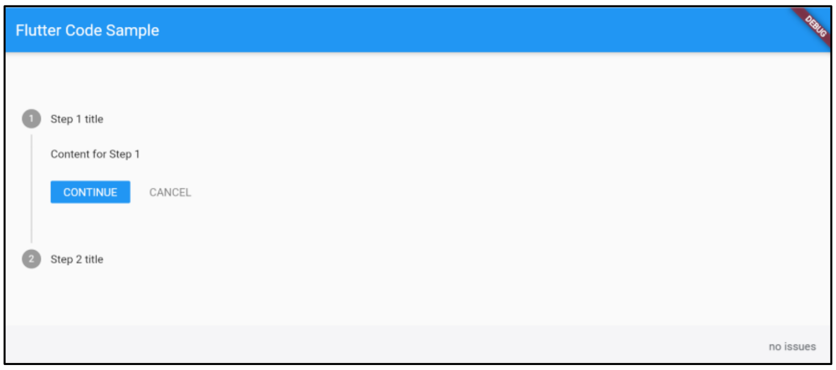


Figure 4.21
Stepper

5. Information displays

- Card

  A Material Design card. A card has slightly rounded corners and a shadow

- Icon

  A Material Design icon.

- Image

  A widget that displays an image.

- LinearProgressIndicator

  A material design linear progress indicator, also known as a progress bar.

- CircularProgressIndicator

  A material design circular progress indicator, which spins to indicate that the application is busy.

4.2 Flutter Architecture

Flutter delivers the basic architecture that can be applied to application and manage its state easily. The architecture that is used in Flutter is called the Business Logic Component (BLOC). It is an event-state based approach that allows you to trigger events and handle state changes based on them. The BLOC is a good approach that separates your business logic from the user interface and oversees business logic key points by testing. The core ideas that were used for BLOC architecture are simplicity, scalability, and testability, and all these goals were definitely achieved within the BLOC architecture. But this is a separate topic that we may cover at a later date.

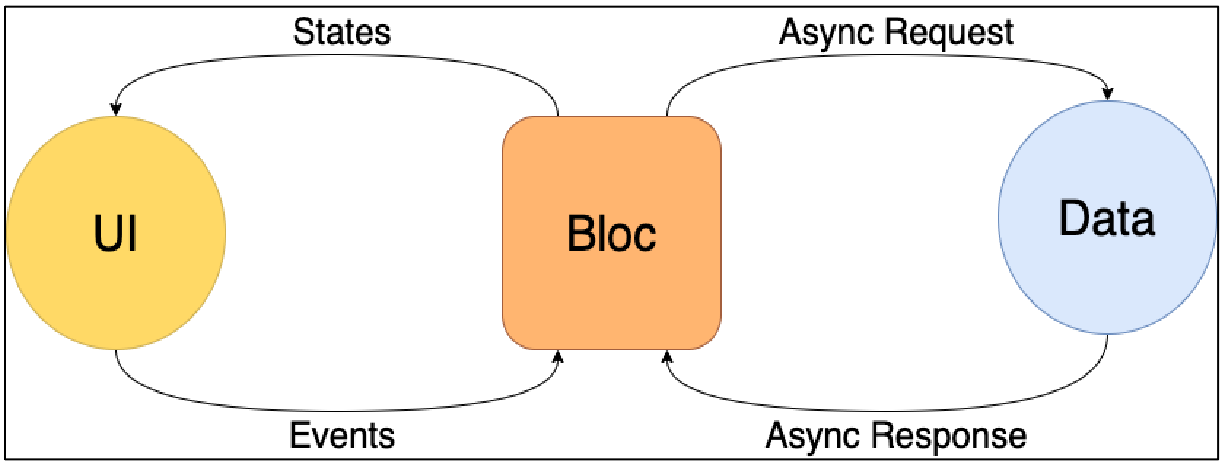


Figure 4.22
Flutter Architecture

5. DETAILED DESIGN

5.1 Process Flow

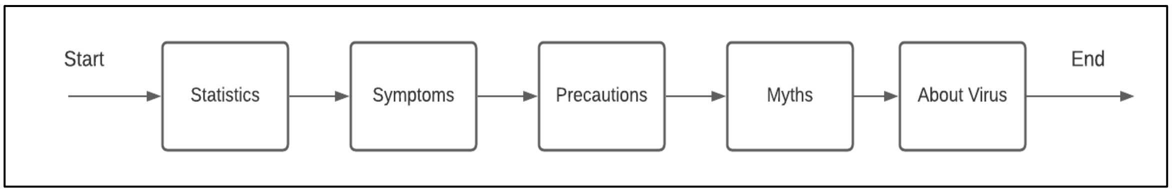


Figure 5.1
Process Flow-1

The user is provided with many options to choose like View the Statistics, read about the Symptoms of COVID-19, read about Precautions to be taken, Myths regarding the virus and more information About the virus. The user can choose any of the options in any random way.

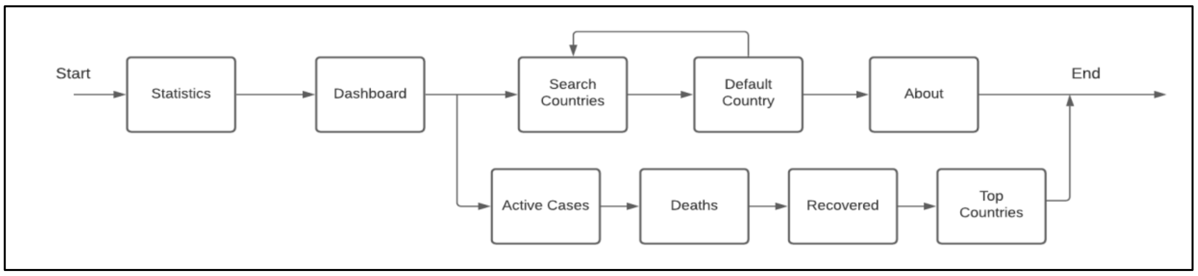


Figure 5.2
Process Flow-2

Statistics screen consists of many components. The user is provided with the information on global statistics of COVID-19 spread. The user can view the number of active cases, deaths and recovered cases of any country from the dashboard. The dashboard also contains top countries with highest number of active cases. The Search feature allows the user to search any country of his choice to view information. The user can choose a default country. When the default country is to be selected it navigates to search screen. About screen provides brief information on what the app does to the user.

5.2 Screen Designs

The screen below shows the main screen of the app. It includes different features like Statistics, Symptoms, Precautions, Myths and About virus represented in card view along with suitable headings and descriptions.

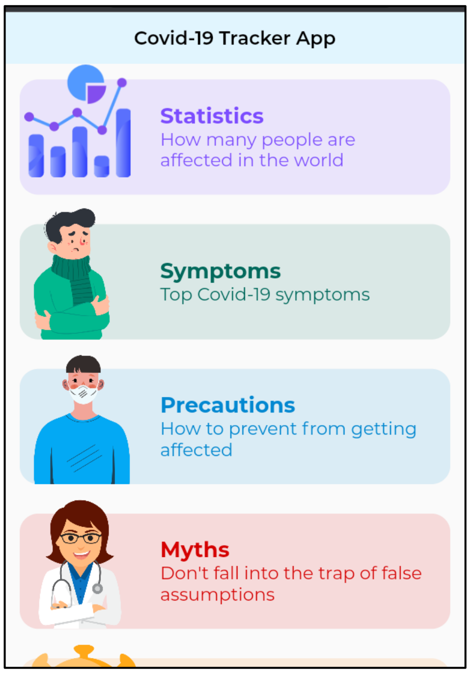

Covid-19 Tracker App
Statistics
How many people are affected in the world
Symptoms
Top Covid-19 symptoms
Precautions
How to prevent from getting affected
Myths
Don't fall into the trap of false assumptions

Figure 5.3
Main Screen

The screen below represents the global statistics screen which includes Bottom Navigation Bar as a dashboard which initially shows the active, deaths and recovered cases along with affected areas in the world map.

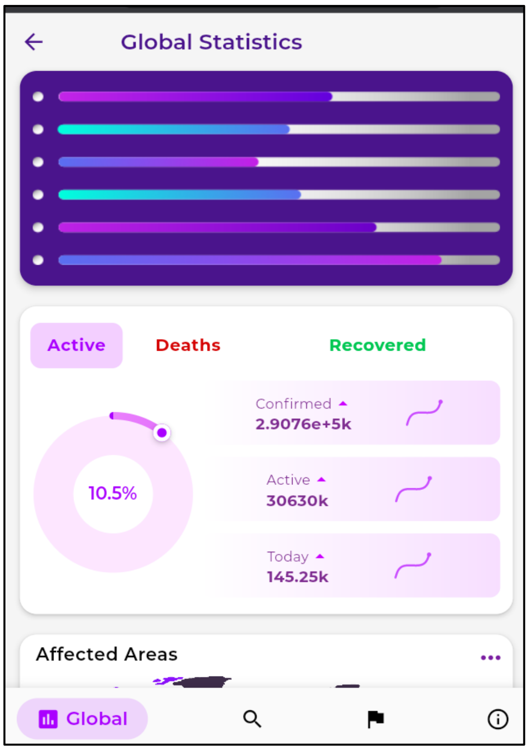


Figure 5.4
Global Statistics Screen

The screen below includes many card views for all the countries along with their flags and a search box is provided to find the desired country.

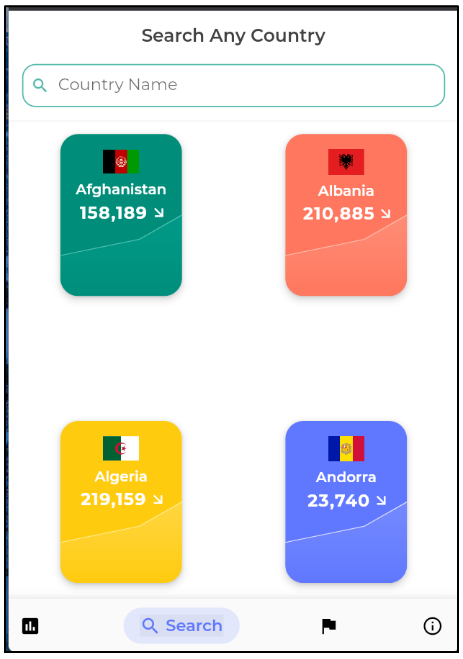


Figure 5.5
Card View of Countries

The screen below helps the user to choose a default country, for immediate updates. It uses button to navigate the user to next screen.

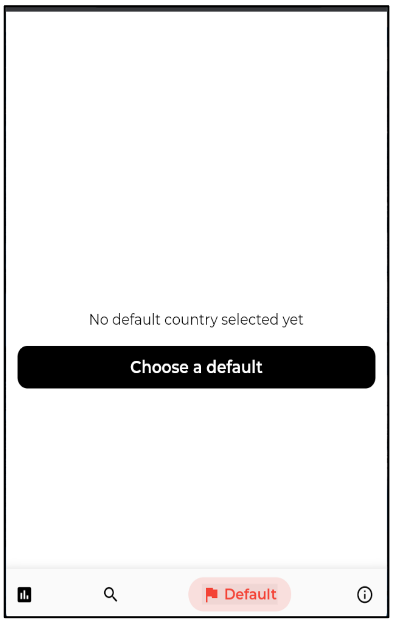
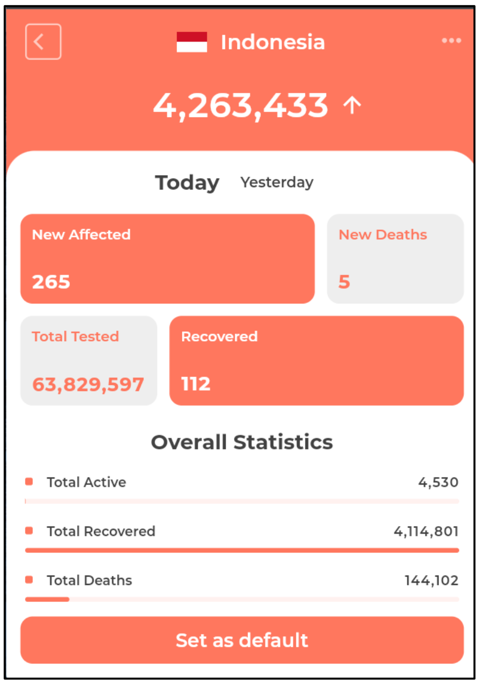


Figure 5.6
Default Country

The screen below is displayed when the user clicks on the button to choose a default country. It shows the user the present and the previous day overall statistics.



Indonesia

4,263,433 ↑

Today Yesterday

New Affected	New Deaths
265	5

Total Tested	Recovered
63,829,597	112

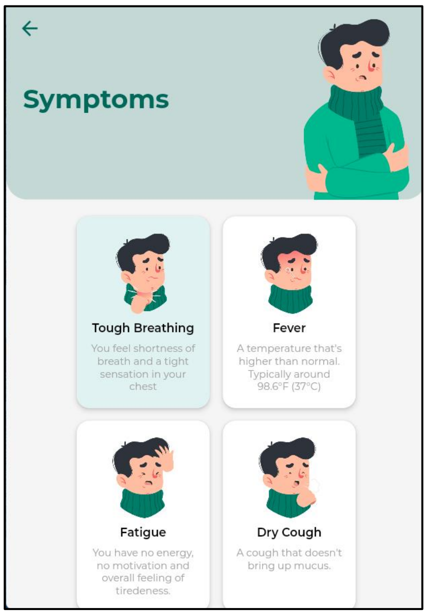
Overall Statistics

Total Active	4,530
Total Recovered	4,114,801
Total Deaths	144,102

Set as default

Figure 5.7
Country Statistics

The screen below provides information on various symptoms of COVID-19 designed using card view and suitable images.



Symptoms

Tough Breathing
You feel shortness of breath and a tight sensation in your chest

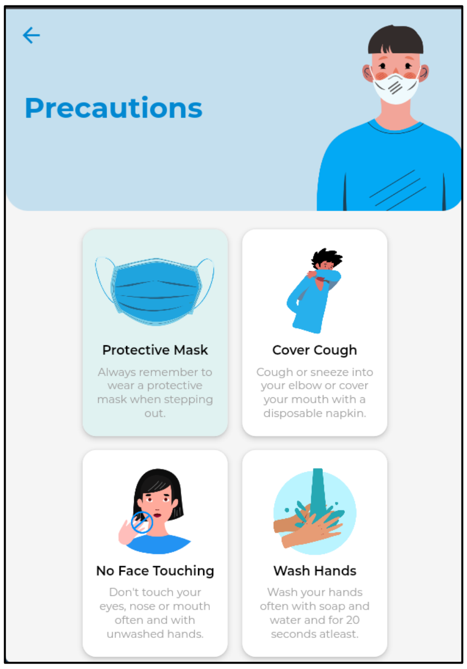
Fever
A temperature that's higher than normal. Typically around 98.6°F (37°C)

Fatigue
You have no energy, no motivation and overall feeling of tiredeness.

Dry Cough
A cough that doesn't bring up mucus.

Figure 5.8
Symptoms

The screen below provides information on various preventive measures to be taken to avoid the spread of COVID-19 and is designed using card view and suitable images.



Precautions

Protective Mask

Always remember to wear a protective mask when stepping out.

Cover Cough

Cough or sneeze into your elbow or cover your mouth with a disposable napkin.

No Face Touching

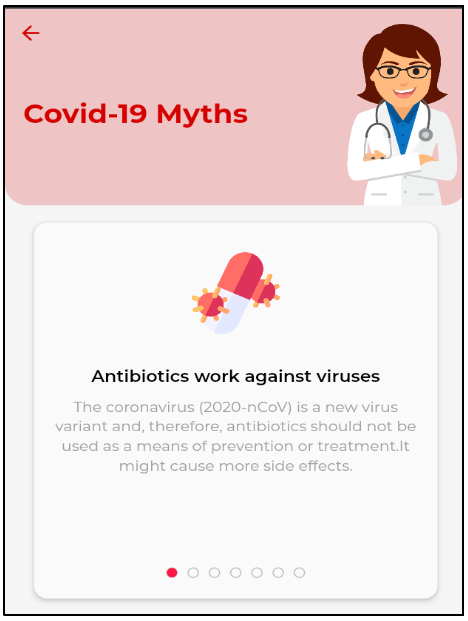
Don't touch your eyes, nose or mouth often and with unwashed hands.

Wash Hands

Wash your hands often with soap and water and for 20 seconds atleast.

Figure 5.9
Precautions

The screen below displays various myths related to COVID-19 and is designed using a feature to swipe cards, present along with description and suitable images.



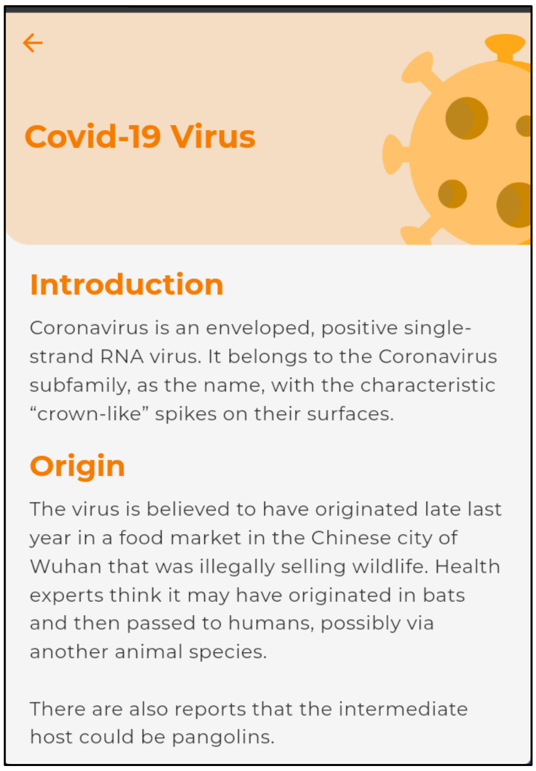
Covid-19 Myths

Antibiotics work against viruses

The coronavirus (2020-nCoV) is a new virus variant and, therefore, antibiotics should not be used as a means of prevention or treatment.It might cause more side effects.

Figure 5.10
Covid-19 Myths

The screen below displays information on what actually COVID-19 is and its history and its effects on human race. It also attempts to provide answers to general questions on the virus.



Covid-19 Virus

Introduction

Coronavirus is an enveloped, positive single-strand RNA virus. It belongs to the Coronavirus subfamily, as the name, with the characteristic "crown-like" spikes on their surfaces.

Origin

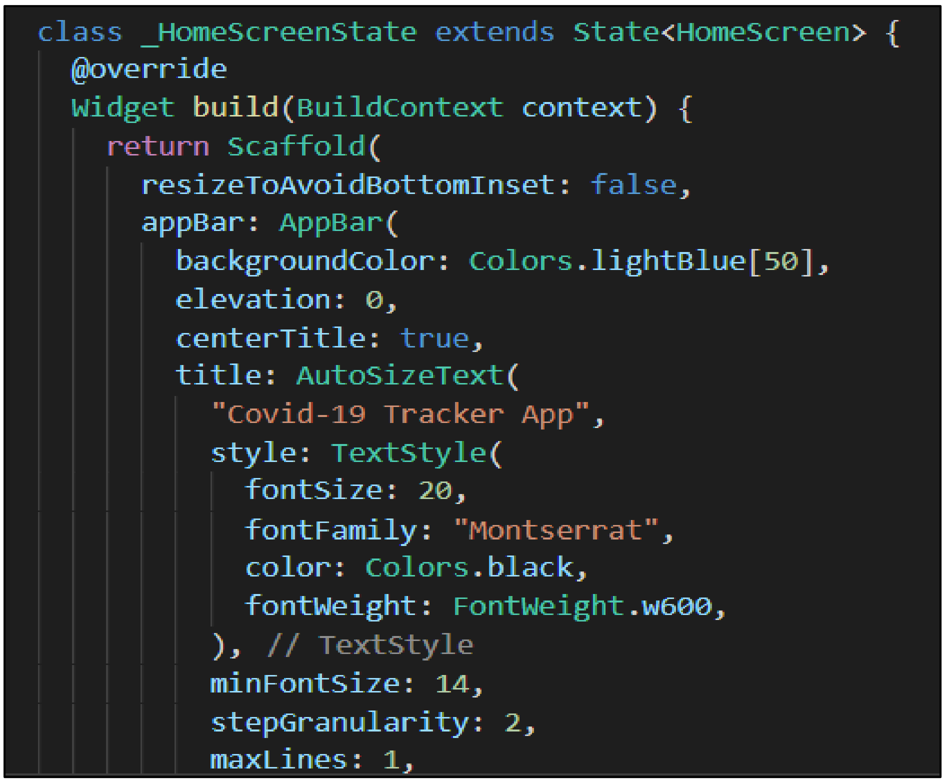
The virus is believed to have originated late last year in a food market in the Chinese city of Wuhan that was illegally selling wildlife. Health experts think it may have originated in bats and then passed to humans, possibly via another animal species.

There are also reports that the intermediate host could be pangolins.

Figure 5.11
Covid-19 Information

6. IMPLEMENTATION DETAILS

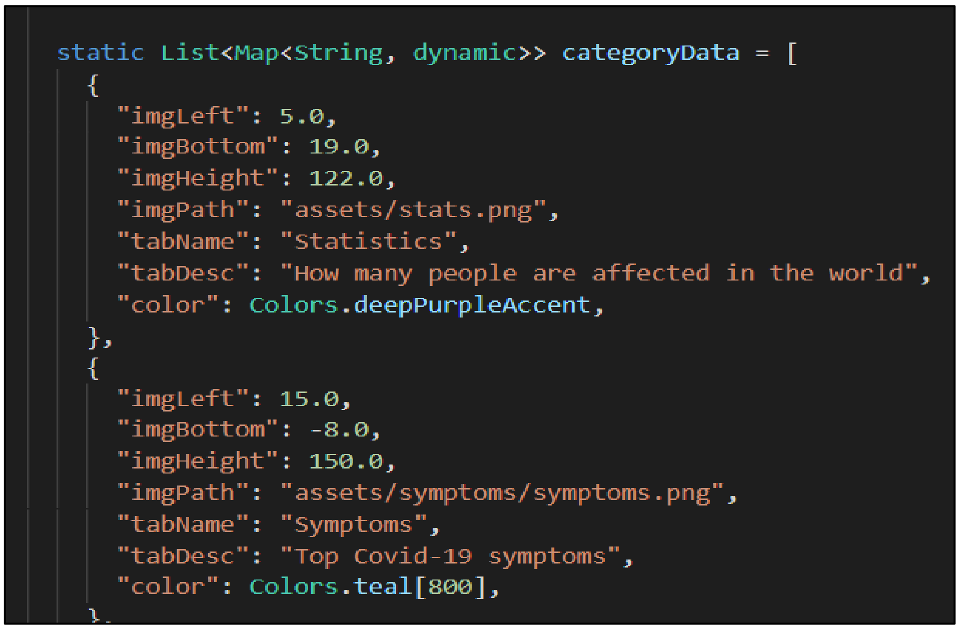
6.1 Codes



```
class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      appBar: AppBar(
        backgroundColor: Colors.lightBlue[50],
        elevation: 0,
        centerTitle: true,
        title: AutoSizeText(
          "Covid-19 Tracker App",
          style: TextStyle(
            fontSize: 20,
            fontFamily: "Montserrat",
            color: Colors.black,
            fontWeight: FontWeight.w600,
          ), // TextStyle
          minFontSize: 14,
          stepGranularity: 2,
          maxLines: 1,
```

Figure 6.1
Home page

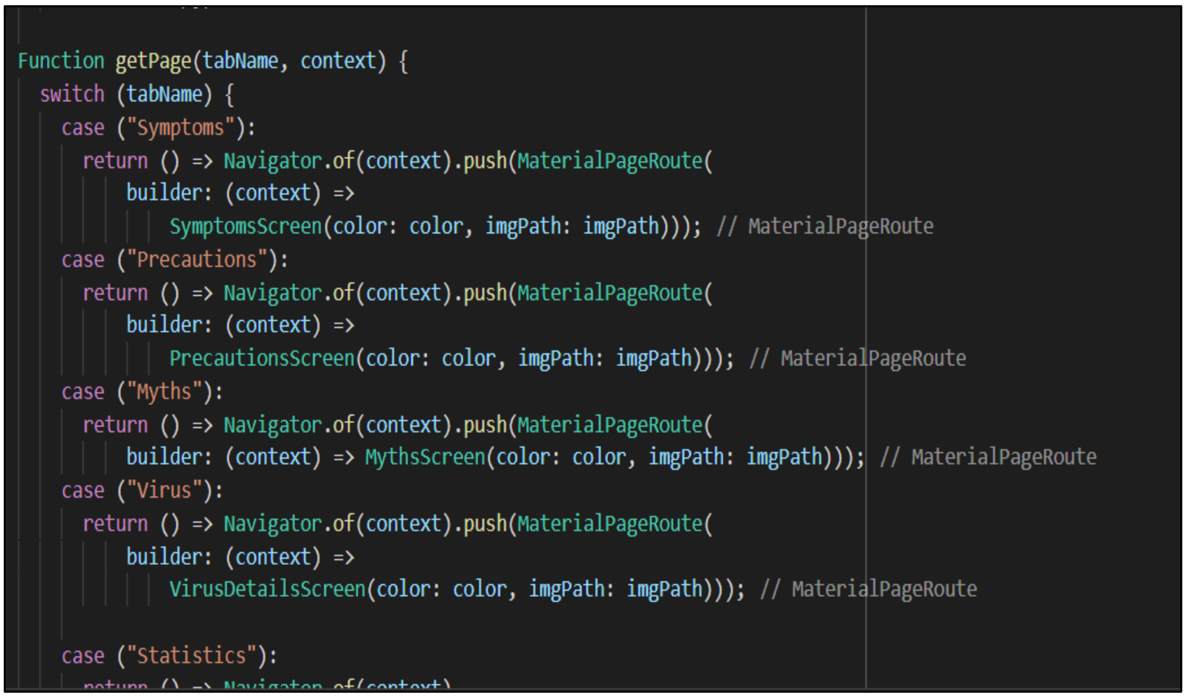
The snippet is used to include basic design to the home page like AppBar, font, colors, and other necessary widgets.



```
static List<Map<String, dynamic>> categoryData = [
  {
    "imgLeft": 5.0,
    "imgBottom": 19.0,
    "imgHeight": 122.0,
    "imgPath": "assets/stats.png",
    "tabName": "Statistics",
    "tabDesc": "How many people are affected in the world",
    "color": Colors.deepPurpleAccent,
  },
  {
    "imgLeft": 15.0,
    "imgBottom": -8.0,
    "imgHeight": 150.0,
    "imgPath": "assets/symptoms/symptoms.png",
    "tabName": "Symptoms",
    "tabDesc": "Top Covid-19 symptoms",
    "color": Colors.teal[800],
```

Figure 6.2
Home Category-1

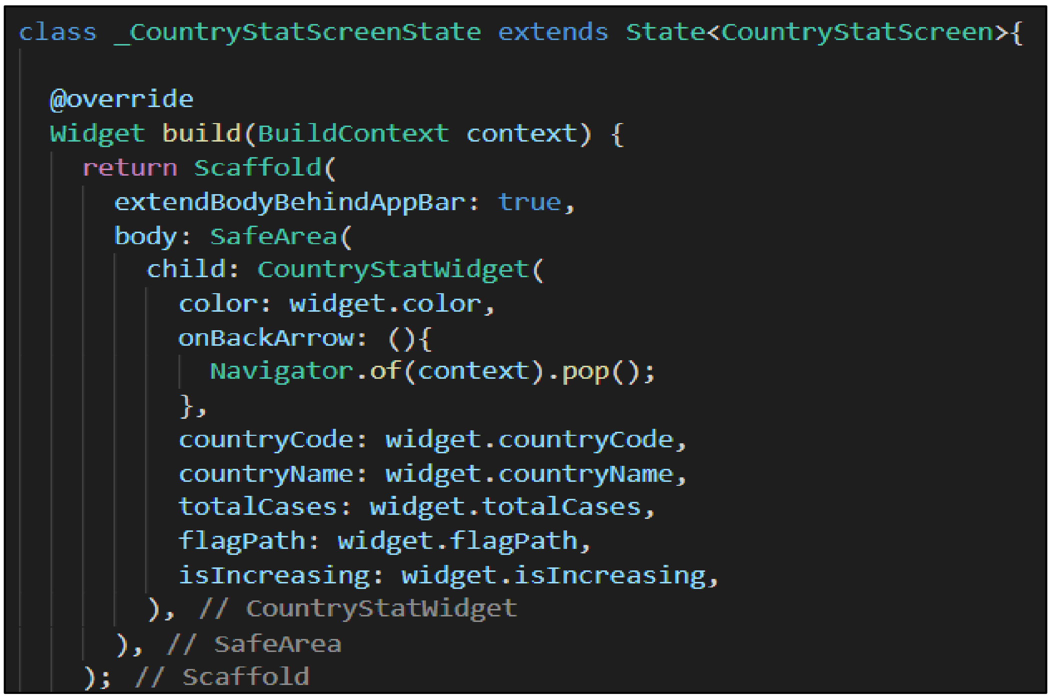
The snippet is used to display different features that the app provides on the home screen.



```
Function getPage(tabName, context) {
  switch (tabName) {
    case ("Symptoms"):
      return () => Navigator.of(context).push(MaterialPageRoute(
          builder: (context) =>
              SymptomsScreen(color: color, imgPath: imgPath))); // MaterialPageRoute
    case ("Precautions"):
      return () => Navigator.of(context).push(MaterialPageRoute(
          builder: (context) =>
              PrecautionsScreen(color: color, imgPath: imgPath))); // MaterialPageRoute
    case ("Myths"):
      return () => Navigator.of(context).push(MaterialPageRoute(
          builder: (context) => MythsScreen(color: color, imgPath: imgPath))); // MaterialPageRoute
    case ("Virus"):
      return () => Navigator.of(context).push(MaterialPageRoute(
          builder: (context) =>
              VirusDetailsScreen(color: color, imgPath: imgPath))); // MaterialPageRoute

    case ("Statistics"):
```

Figure 6.3
Home Category-2

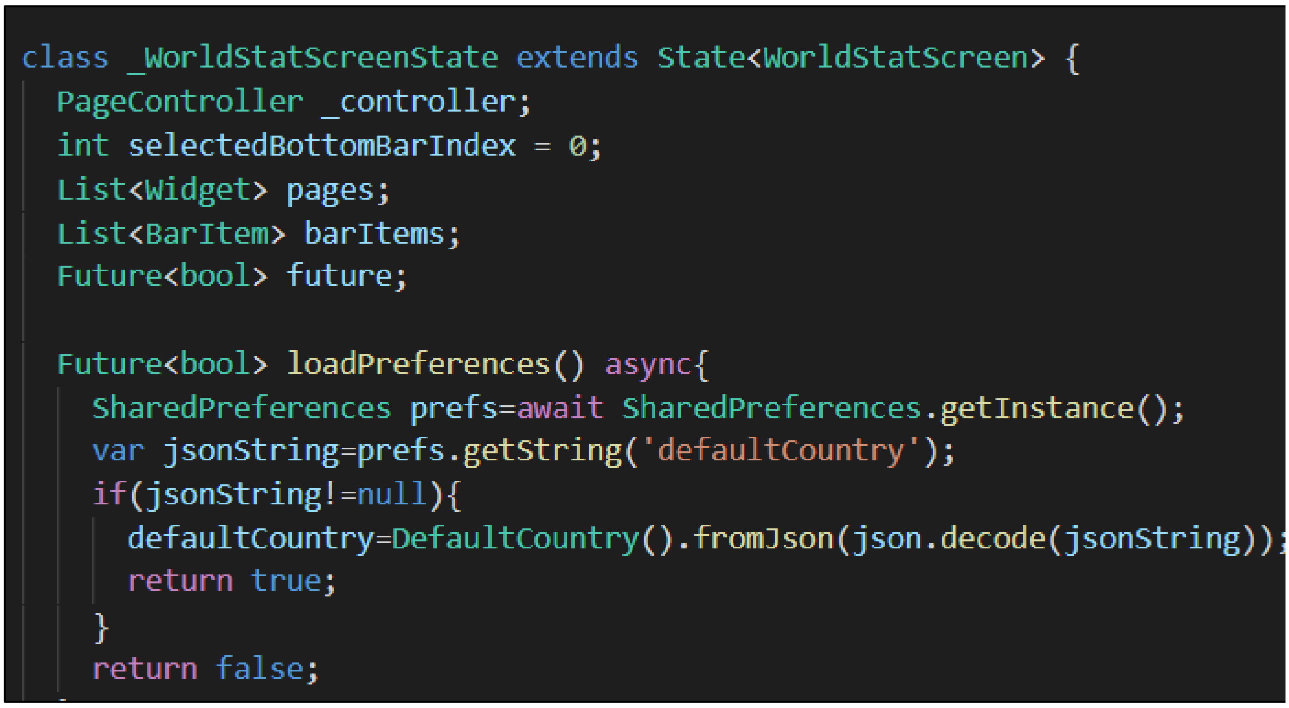
The snippet is used to switch the tabs and fetch the specified screens, when a particular card is selected.'



```
class _CountryStatScreenState extends State<CountryStatScreen>{

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBodyBehindAppBar: true,
      body: SafeArea(
        child: CountryStatWidget(
          color: widget.color,
          onBackArrow: (){
            Navigator.of(context).pop();
          },
          countryCode: widget.countryCode,
          countryName: widget.countryName,
          totalCases: widget.totalCases,
          flagPath: widget.flagPath,
          isIncreasing: widget.isIncreasing,
        ), // CountryStatWidget
      ), // SafeArea
    ); // Scaffold
```

Figure 6.4
Statistics Count

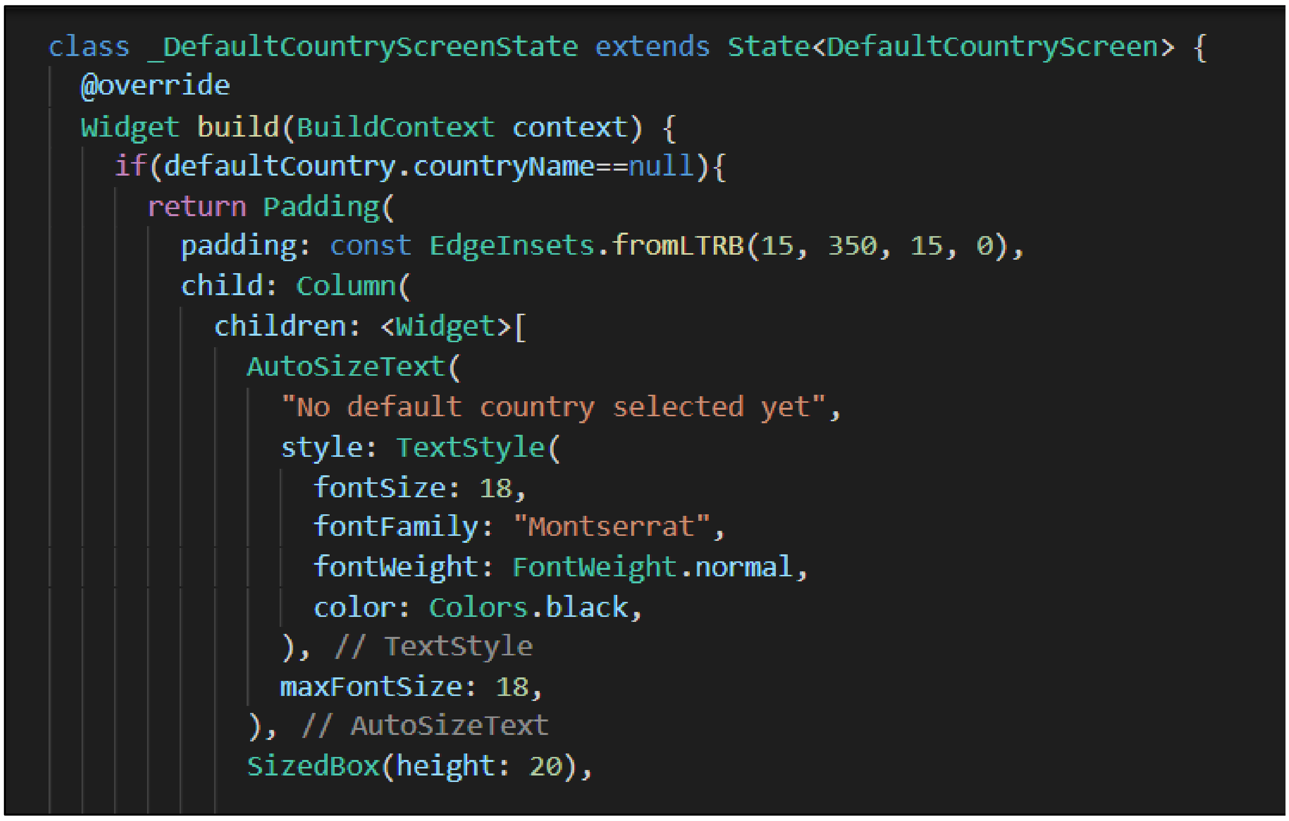
The snippet is implemented to display different statistics of the country and uses mainly Scaffold.



```
class _WorldStatScreenState extends State<WorldStatScreen> {
  PageController _controller;
  int selectedBottomBarIndex = 0;
  List<Widget> pages;
  List<BarItem> barItems;
  Future<bool> future;

  Future<bool> loadPreferences() async{
    SharedPreferences prefs=await SharedPreferences.getInstance();
    var jsonString=prefs.getString('defaultCountry');
    if(jsonString!=null){
      defaultCountry=DefaultCountry().fromJson(json.decode(jsonString));
      return true;
    }
    return false;
```

Figure 6.5
World Statistics

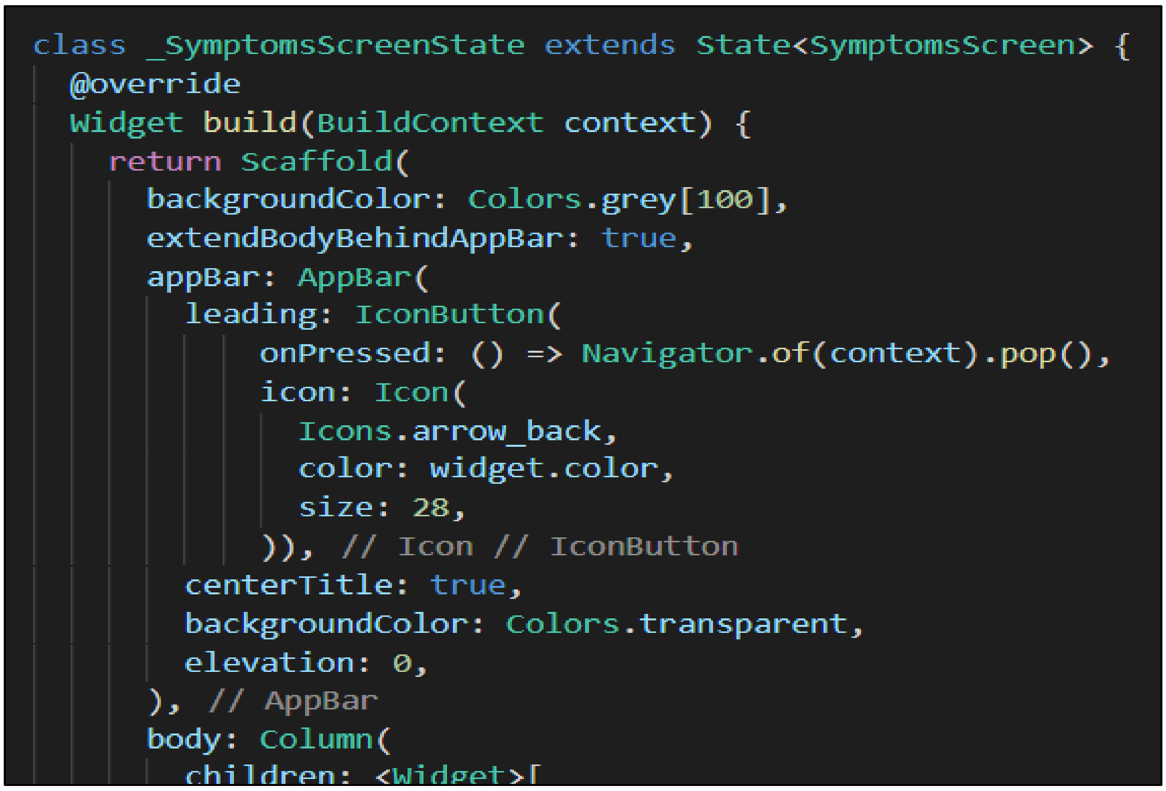
The snippet uses page controller to choose the initial default country.



```
class _DefaultCountryScreenState extends State<DefaultCountryScreen> {
  @override
  Widget build(BuildContext context) {
    if(defaultCountry.countryName==null){
      return Padding(
        padding: const EdgeInsets.fromLTRB(15, 350, 15, 0),
        child: Column(
          children: <Widget>[
            AutoSizeText(
              "No default country selected yet",
              style: TextStyle(
                fontSize: 18,
                fontFamily: "Montserrat",
                fontWeight: FontWeight.normal,
                color: Colors.black,
              ), // TextStyle
              maxFontSize: 18,
            ), // AutoSizeText
            SizedBox(height: 20),
```

Figure 6.6
Default Count

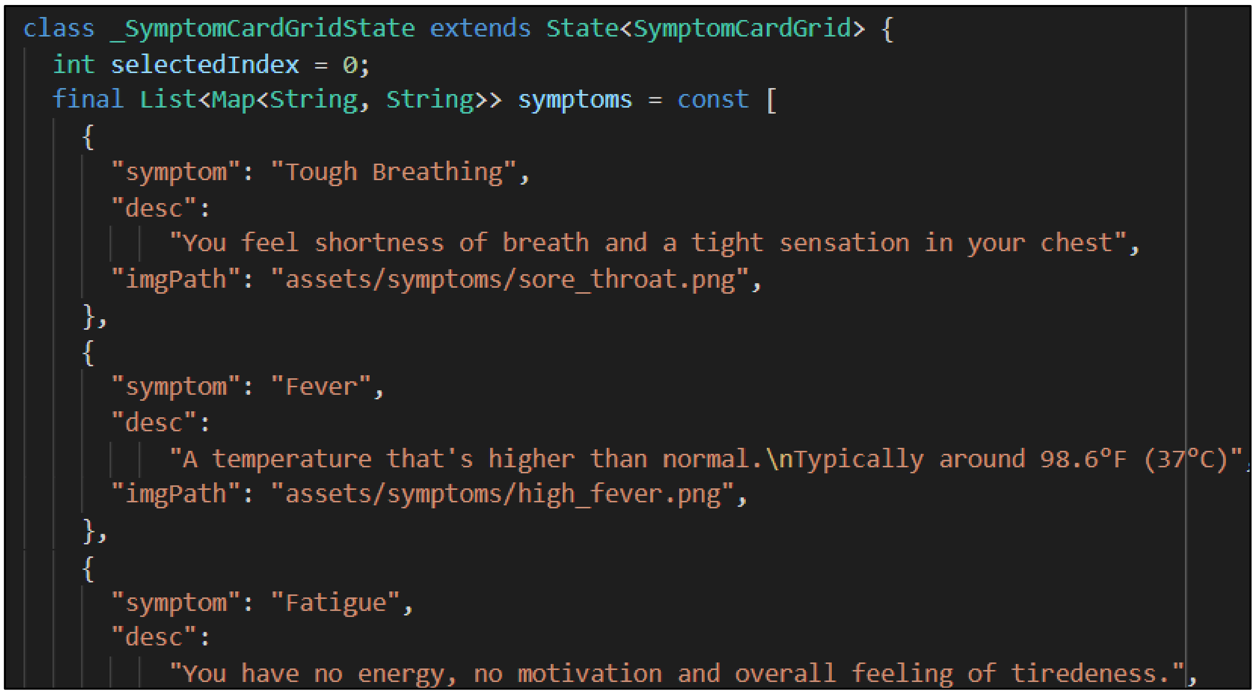
The snippet implements the default country page and displays the message “No default country selected yet” before selection.



```
class _SymptomsScreenState extends State<SymptomsScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      extendBodyBehindAppBar: true,
      appBar: AppBar(
        leading: IconButton(
            onPressed: () => Navigator.of(context).pop(),
            icon: Icon(
              Icons.arrow_back,
              color: widget.color,
              size: 28,
            )), // Icon // IconButton
        centerTitle: true,
        backgroundColor: Colors.transparent,
        elevation: 0,
      ), // AppBar
      body: Column(
        children: <Widget>[
```

Figure 6.7
Symptoms Page

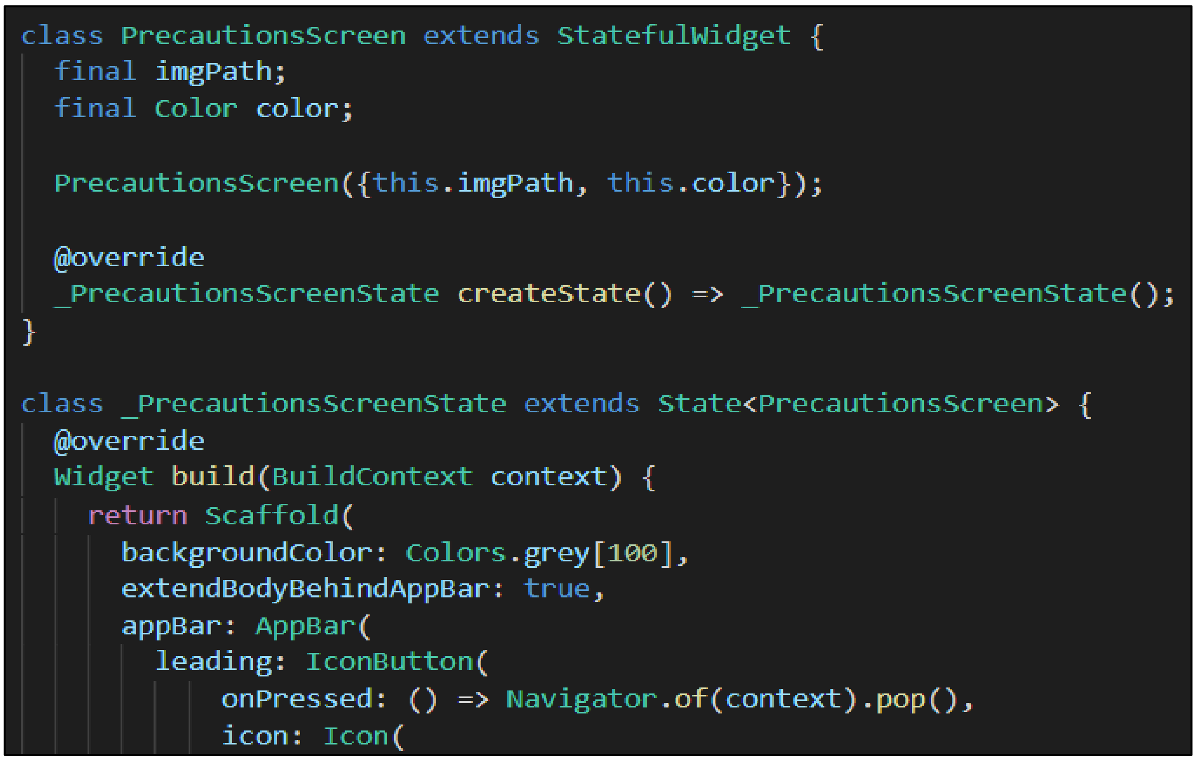
The snippet is used to design the screen using AppBar and Column components to display the necessary information.



```
class _SymptomCardGridState extends State<SymptomCardGrid> {
  int selectedIndex = 0;
  final List<Map<String, String>> symptoms = const [
    {
      "symptom": "Tough Breathing",
      "desc":
          "You feel shortness of breath and a tight sensation in your chest",
      "imgPath": "assets/symptoms/sore_throat.png",
    },
    {
      "symptom": "Fever",
      "desc":
          "A temperature that's higher than normal.\nTypically around 98.6°F (37°C)",
      "imgPath": "assets/symptoms/high_fever.png",
    },
    {
      "symptom": "Fatigue",
      "desc":
          "You have no energy, no motivation and overall feeling of tiredeness.",
```

Figure 6.8
Symptoms

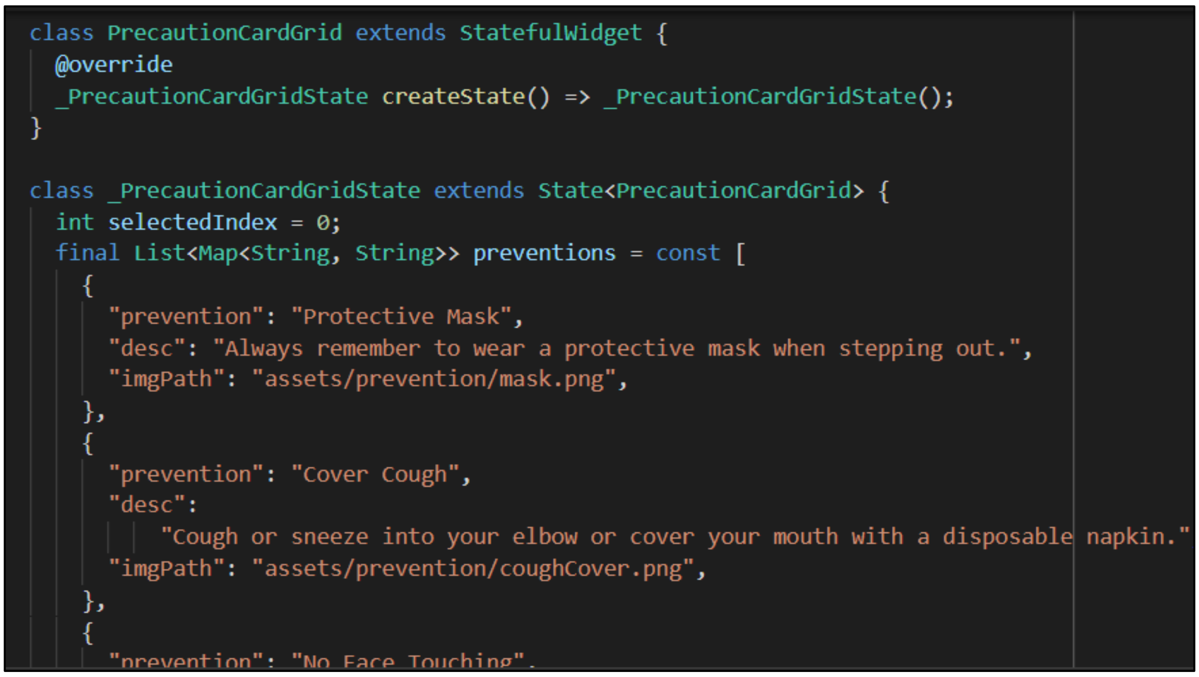
The snippet implements list to store the information that is to be displayed to the user.



```
class PrecautionsScreen extends StatefulWidget {
  final imgPath;
  final Color color;

  PrecautionsScreen({this.imgPath, this.color});

  @override
  _PrecautionsScreenState createState() => _PrecautionsScreenState();
}

class _PrecautionsScreenState extends State<PrecautionsScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      extendBodyBehindAppBar: true,
      appBar: AppBar(
        leading: IconButton(
          onPressed: () => Navigator.of(context).pop(),
          icon: Icon(
```

Figure 6.9
Precautions Screen

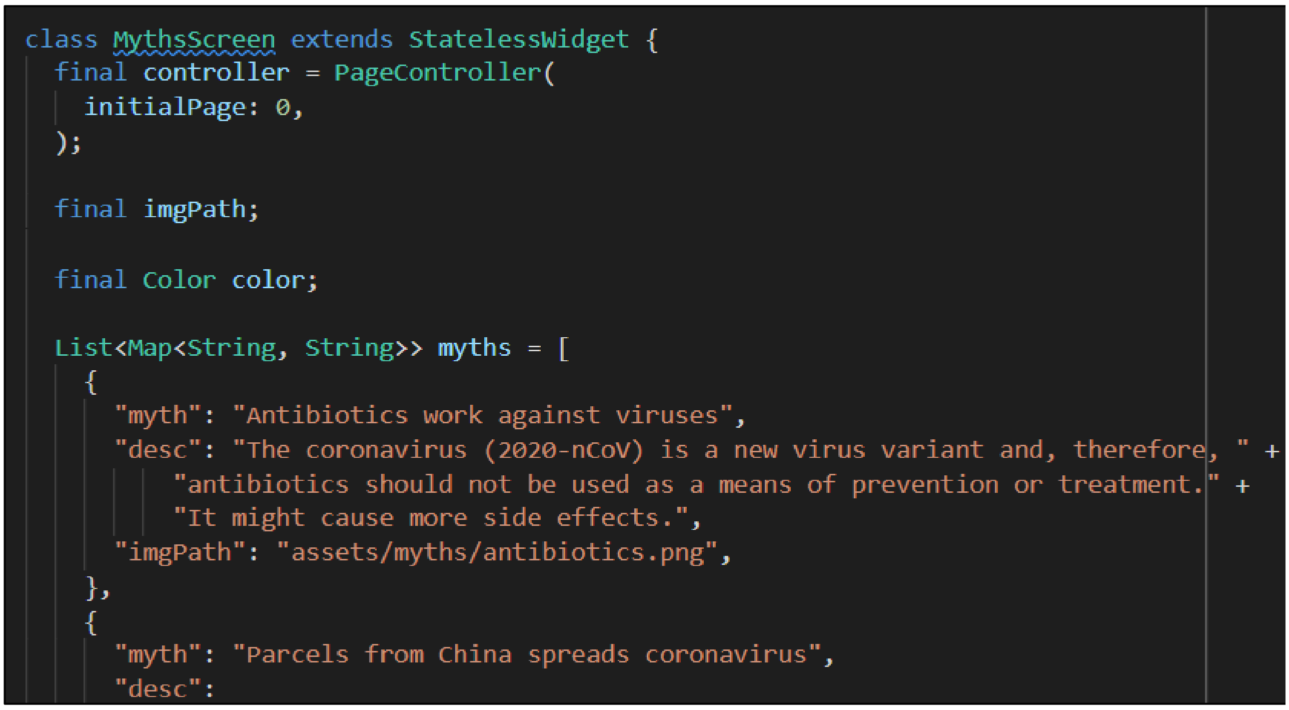
The snippet to design the screen using AppBar and Column components to display the necessary information to prevent the spread of the virus.



```
class PrecautionCardGrid extends StatefulWidget {
  @override
  _PrecautionCardGridState createState() => _PrecautionCardGridState();
}

class _PrecautionCardGridState extends State<PrecautionCardGrid> {
  int selectedIndex = 0;
  final List<Map<String, String>> preventions = const [
    {
      "prevention": "Protective Mask",
      "desc": "Always remember to wear a protective mask when stepping out.",
      "imgPath": "assets/prevention/mask.png",
    },
    {
      "prevention": "Cover Cough",
      "desc":
          "Cough or sneeze into your elbow or cover your mouth with a disposable napkin."
      "imgPath": "assets/prevention/coughCover.png",
    },
    {
      "prevention": "No Face Touching",
```

Figure 6.10
Precautions

The snippet implements list to store the information that is to be displayed to the user in an effective way.



```
class MythsScreen extends StatelessWidget {
  final controller = PageController(
    initialPage: 0,
  );

  final imgPath;

  final Color color;

  List<Map<String, String>> myths = [
    {
      "myth": "Antibiotics work against viruses",
      "desc": "The coronavirus (2020-nCoV) is a new virus variant and, therefore, " +
          "antibiotics should not be used as a means of prevention or treatment." +
          "It might cause more side effects.",
      "imgPath": "assets/myths/antibiotics.png",
    },
    {
      "myth": "Parcels from China spreads coronavirus",
      "desc":
```

Figure 6.11
Myths Screen

The snippet initializes the first page using page controller and implements list to store the information that is to be displayed to the user.

6.2 Implementing Rest API in Flutter

Flutter uses HTTP package, which provides advanced methods to perform operations. REST API uses simple http calls to communicate with JSON data because:

- It uses await & async features.
- It provides various methods.
- It provides class and http to perform web requests.

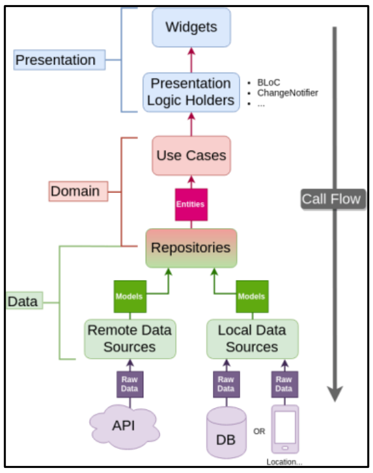


Figure 6.12
HTTP Package

The detail explanation of the core methods of the http package are as follows:

1. Read:

   This method is used to read or retrieve the representation of resources. It requests the specified url by using the get method and returns the response as Future<String>.
2. Get:

   This method requests the specified URL from the get method and returns a response as Future<response>. Here, the response is a class, which holds the response information.
3. Post:

   This method is used to submit the data to the specified resources. It requests the specified url by posting the given data and return a response as Future<response>.
4. Put:

   This method is utilized for update capabilities. It updates all the current representation of the target resource with the request payloads. This method requests the specified URL and returns a response as Future<response>.
5. Head:

   It is similar to the Get method, but without the response body.
6. Delete:

   This method is used to remove all the specified resources.

To fetch data from the internet, these necessary steps are to be followed:

Step 1: Install the latest http package and add it to the project.

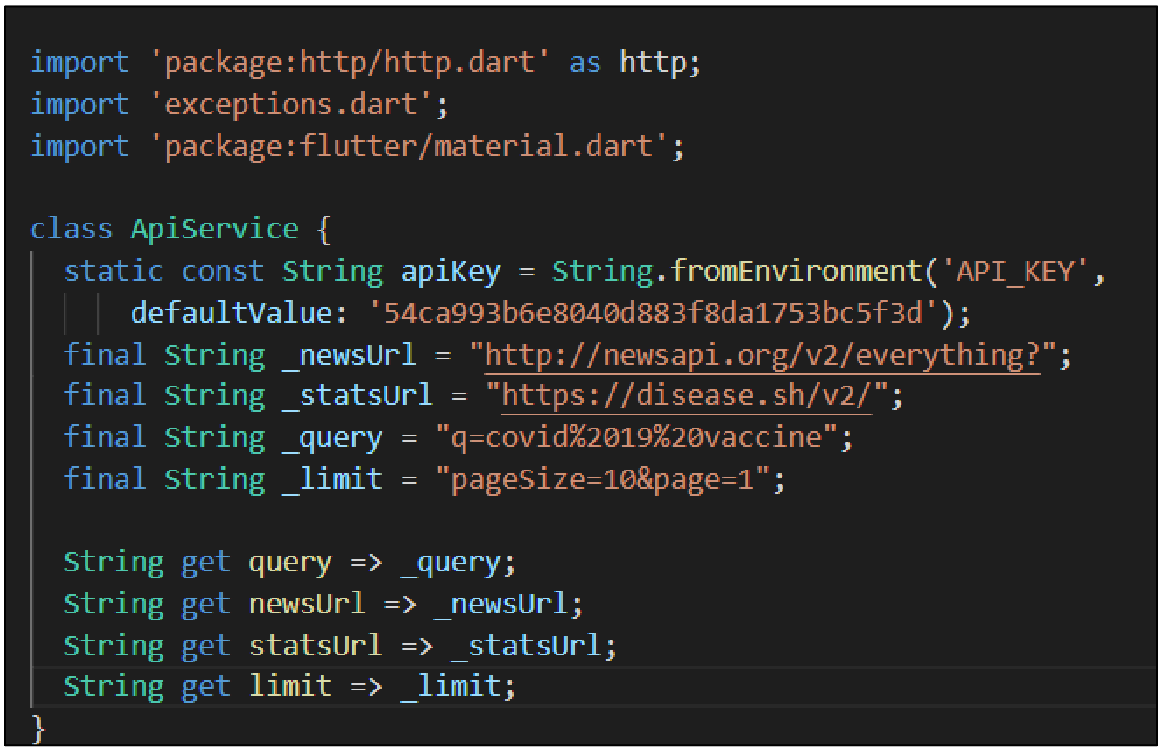
Step 2: Next, make a network request by using the http package.

Step 3: Now, convert the response getting from network request into a custom Dart object.

Step 4: Now, fetch the data with Flutter. You can call the fetch method in the initState(). The following code explains how you can fetch the data.

Step 5: Finally, display the data. You can display the data by using the Future Builder widget. This widget can work easily with async data sources.

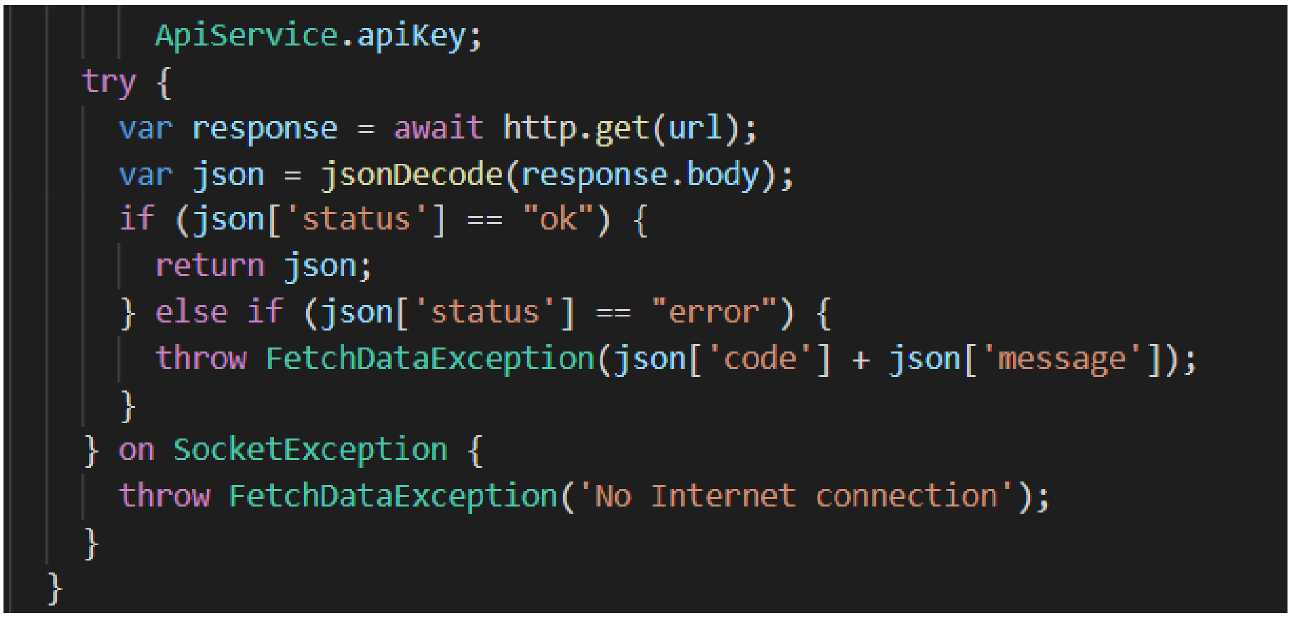
The application uses the API from open-source website https://disease.sh/v2/ . The features of disease.sh - Open Disease Data API is that we can build anything from console widgets to mobile applications, freely and is easy to use. The API provides data on current global outbreaks, including COVID-19 and Influenza.



```dart
import 'package:http/http.dart' as http;
import 'exceptions.dart';
import 'package:flutter/material.dart';

class ApiService {
  static const String apiKey = String.fromEnvironment('API_KEY',
      defaultValue: '54ca993b6e8040d883f8da1753bc5f3d');
  final String _newsUrl = "http://newsapi.org/v2/everything?";
  final String _statsUrl = "https://disease.sh/v2/";
  final String _query = "q=covid%2019%20vaccine";
  final String _limit = "pageSize=10&page=1";

  String get query => _query;
  String get newsUrl => _newsUrl;
  String get statsUrl => _statsUrl;
  String get limit => _limit;
}
```

Figure 6.13
App Client-1

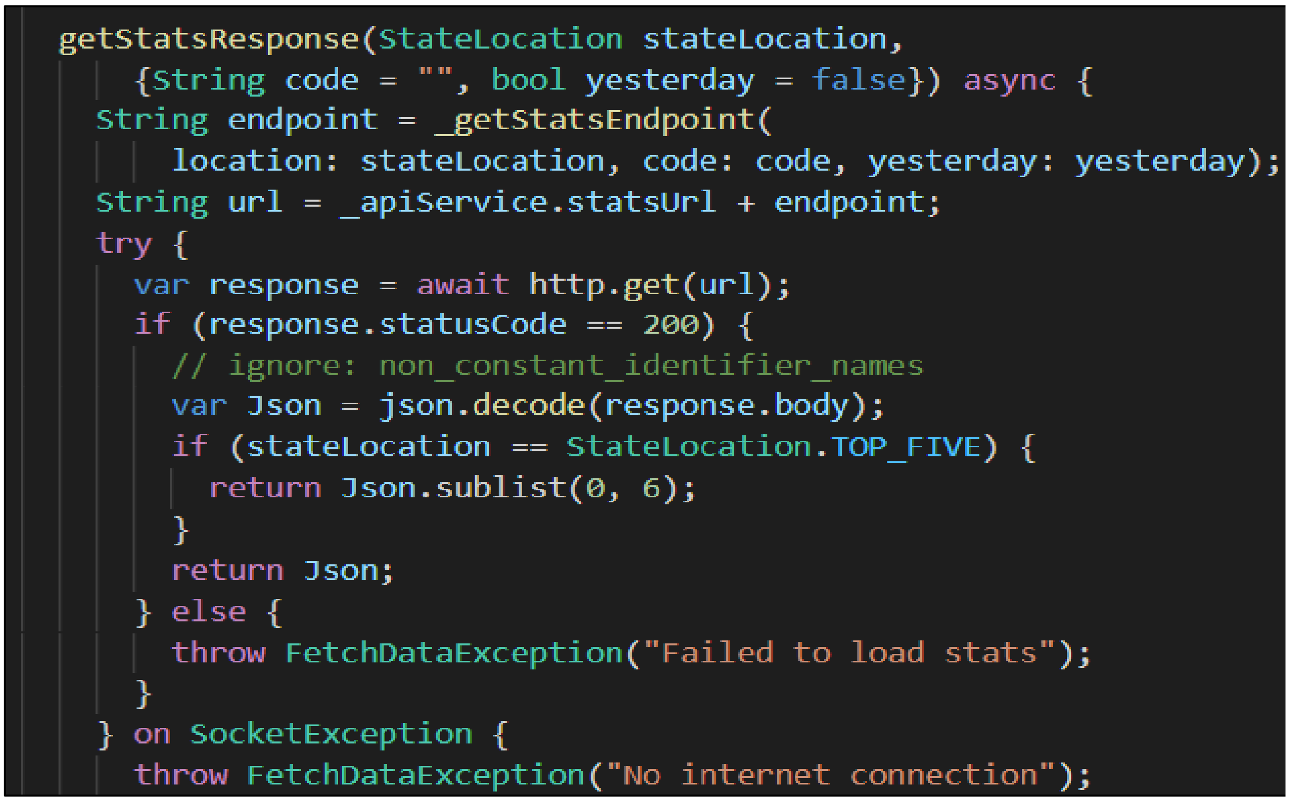
The snippet shows the API Key along with the URL which provide the service.



```dart
        ApiService.apiKey;
    try {
      var response = await http.get(url);
      var json = jsonDecode(response.body);
      if (json['status'] == "ok") {
        return json;
      } else if (json['status'] == "error") {
        throw FetchDataException(json['code'] + json['message']);
      }
    } on SocketException {
      throw FetchDataException('No Internet connection');
    }
  }
```

Figure 6.14
App Client-2

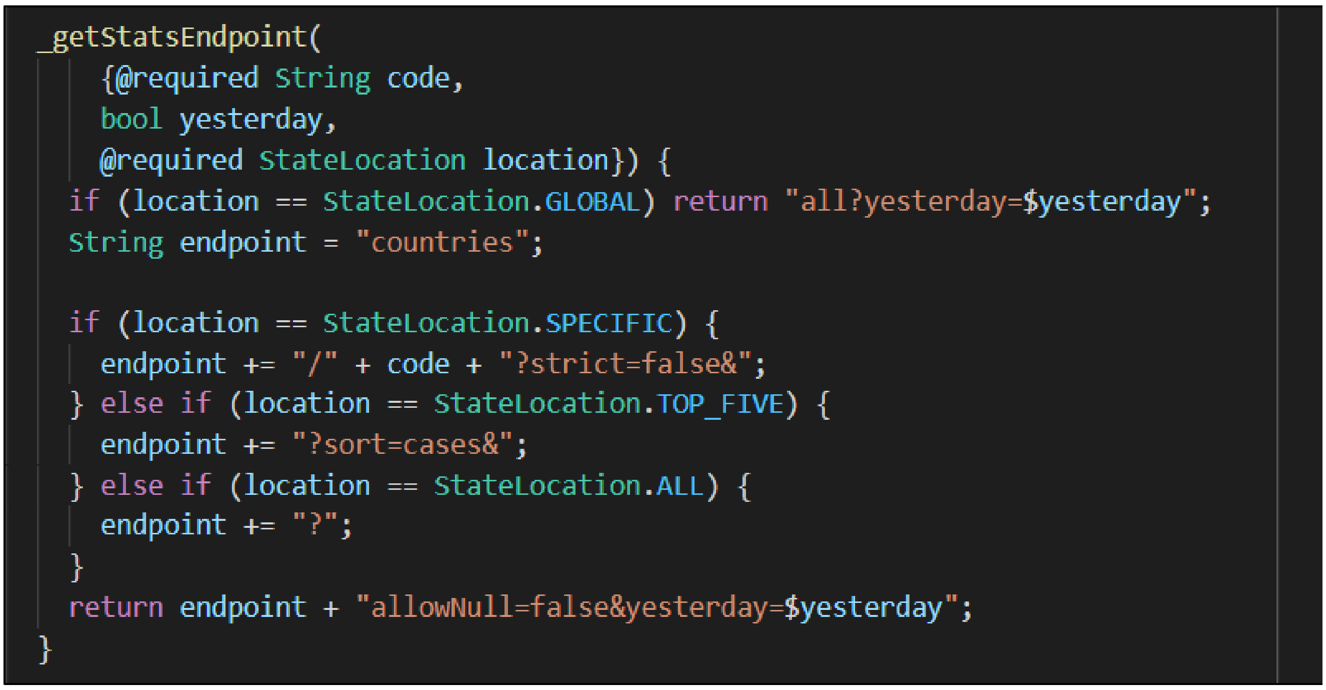
The snippet converts the http.response to a Post. It checks for the necessary conditions and throws exceptions if the errors are found.



```
getStatsResponse(StateLocation stateLocation,
    {String code = "", bool yesterday = false}) async {
  String endpoint = _getStatsEndpoint(
      location: stateLocation, code: code, yesterday: yesterday);
  String url = _apiService.statsUrl + endpoint;
  try {
    var response = await http.get(url);
    if (response.statusCode == 200) {
      // ignore: non_constant_identifier_names
      var Json = json.decode(response.body);
      if (stateLocation == StateLocation.TOP_FIVE) {
        return Json.sublist(0, 6);
      }
      return Json;
    } else {
      throw FetchDataException("Failed to load stats");
    }
  } on SocketException {
    throw FetchDataException("No internet connection");
```

Figure 6.15
App Client-3

The snippet checks for the response received. Based on the status code received, appropriate actions are performed.



```
_getStatsEndpoint(
    {@required String code,
    bool yesterday,
    @required StateLocation location}) {
  if (location == StateLocation.GLOBAL) return "all?yesterday=$yesterday";
  String endpoint = "countries";

  if (location == StateLocation.SPECIFIC) {
    endpoint += "/" + code + "?strict=false&";
  } else if (location == StateLocation.TOP_FIVE) {
    endpoint += "?sort=cases&";
  } else if (location == StateLocation.ALL) {
    endpoint += "?";
  }
  return endpoint + "allowNull=false&yesterday=$yesterday";
}
```

Figure 6.16
App Client-4

The snippet uses the data received from the API to display the contents on the screen by filtering out the necessary information.

7. TESTING

Testing is the process of evaluating and verifying that a software product or application does what it is supposed to do.

7.1 Common Flutter Errors

- <u>A Render-Flex overflowed:</u>

  Render-Flex overflow is one of the most frequently encountered Flutter framework errors. When it happens, you'll see yellow & black stripes indicating the area of overflow in the app UI. The error often occurs when a Column or Row has a child widget that is not constrained in its size.

  How to fix it?

  One way to do it is to wrap the Column in an Expanded widget. Another way is to wrap the Column in a Flexible widget and specify a flex factor. In fact, the Expanded widget is equivalent to the Flexible widget with a flex factor of 1.0.

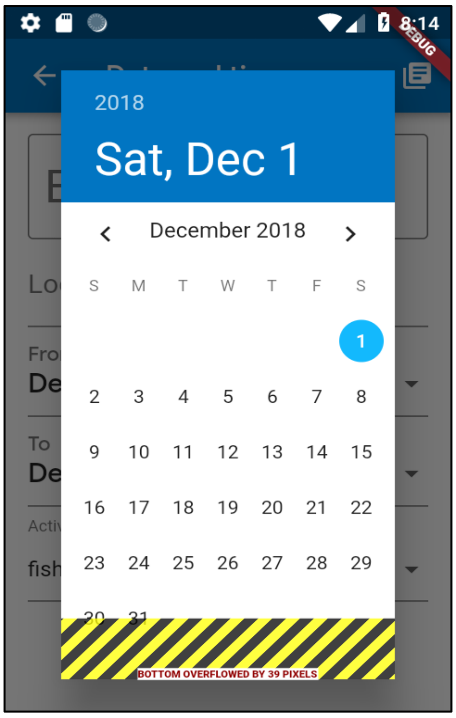


Figure 7.1
Render-Flex overflowed Error

- <u>Render-Box was not laid out:</u>

  While this error is pretty common, it's often a side effect of a primary error occurring earlier in the rendering pipeline. the issue is related to violation of box constraints, and it needs to be solved by providing more information to Flutter about how you'd like to constrain the widgets in question. The Render-Box was not laid out error is often caused by one of two other errors:

  - 'Vertical viewport was given unbounded height'
  - 'An Input-Decorator…cannot have an unbounded width'

- Vertical viewport was given unbounded height:

  This is another common layout error you could run into while creating a UI in Flutter app. The error is often caused when a ListView (or other kinds of scrollable widgets such as GridView) is placed inside a Column. A ListView takes all the vertical space available to it, unless it's constrained by its parent widget. However, a Column doesn't impose any constraint on its children's height by default. The combination of the two behaviors leads to the failure of determining the size of the ListView.

  How to fix it?

  To fix this error, specify how tall the ListView should be. To make it as tall as the remaining space in the Column, wrap it using an Expanded widget. Otherwise, specify an absolute height using a SizedBox widget or a relative height using a Flexible widget.

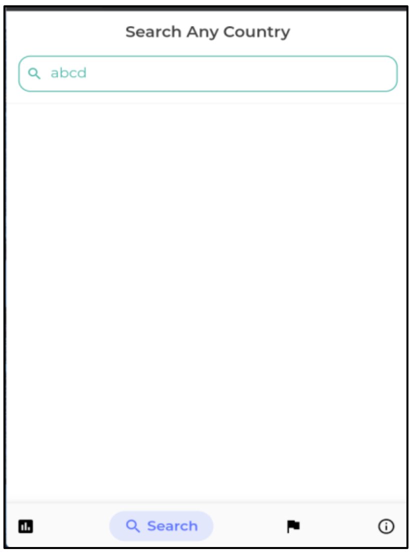


Figure 7.2
Vertical Viewport Error-1

The screen displays the scenario when the user enters a non-existing country name. As we expect, no country is displayed until the correct input is provided.

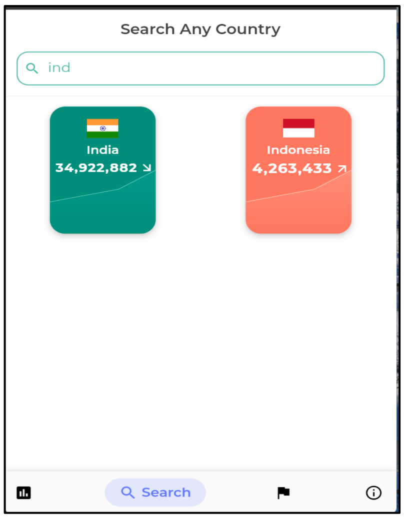


Figure 7.3
Vertical Viewport Error-2

The screen represents the scenario when the user has entered few letters in the search bar and the countries with name matching to the entered value is displayed to the user to be selected.

7.2 Handling Errors in REST API

The simplest way we handle errors is to respond with an appropriate status code.

Here are some common response codes:

400 Bad Request – client sent an invalid request, such as lacking required request body or parameter

401 Unauthorized – client failed to authenticate with the server

403 Forbidden – client authenticated but does not have permission to access the requested resource

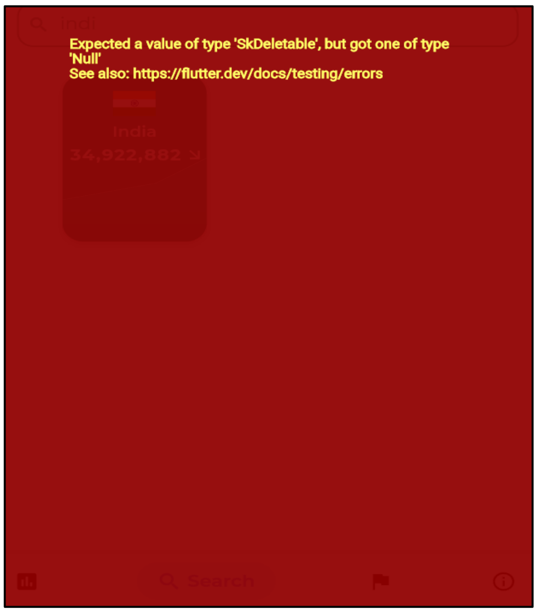
404 Not Found – the requested resource does not exist

412 Precondition Failed – one or more conditions in the request header fields evaluated to false

500 Internal Server Error – a generic error occurred on the server

503 Service Unavailable – the requested service is not available

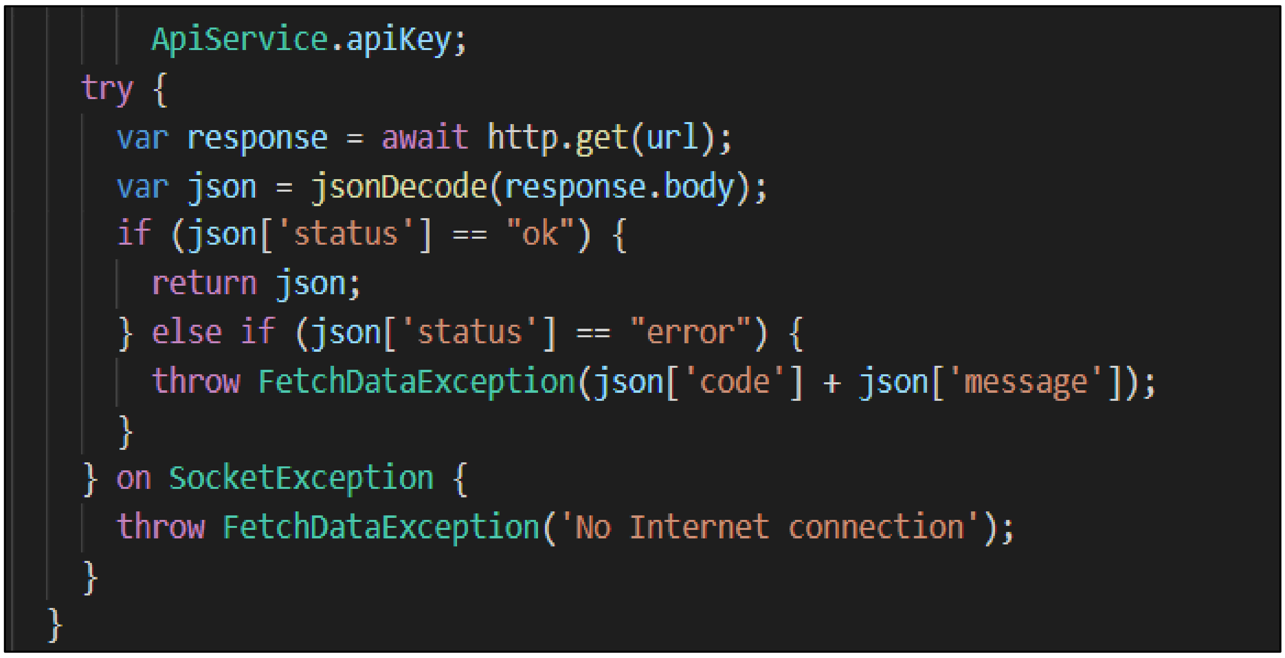
To minimize these kinds of responses to the client, we should diligently attempt to handle or catch internal errors and respond with other appropriate status codes wherever possible.



```
Expected a value of type 'SkDeletable', but got one of type
'Null'
See also: https://flutter.dev/docs/testing/errors
```

Figure 7.4
Type-Error

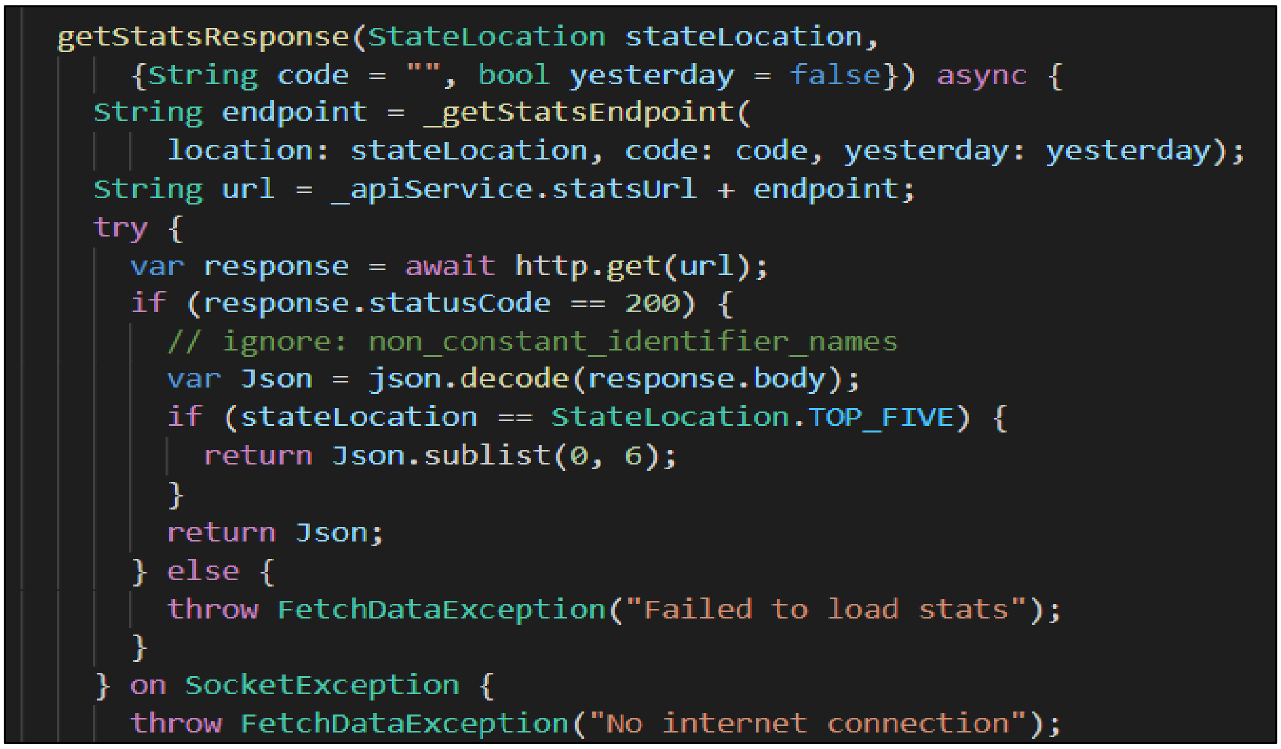
The screen displays a case when the value expected is different from the one that is received from the API (generally null value). Best practice is to check if the variable is null before generating



```
        ApiService.apiKey;
    try {
      var response = await http.get(url);
      var json = jsonDecode(response.body);
      if (json['status'] == "ok") {
        return json;
      } else if (json['status'] == "error") {
        throw FetchDataException(json['code'] + json['message']);
      }
    } on SocketException {
      throw FetchDataException('No Internet connection');
    }
  }
```

Figure 7.5
App Client-1

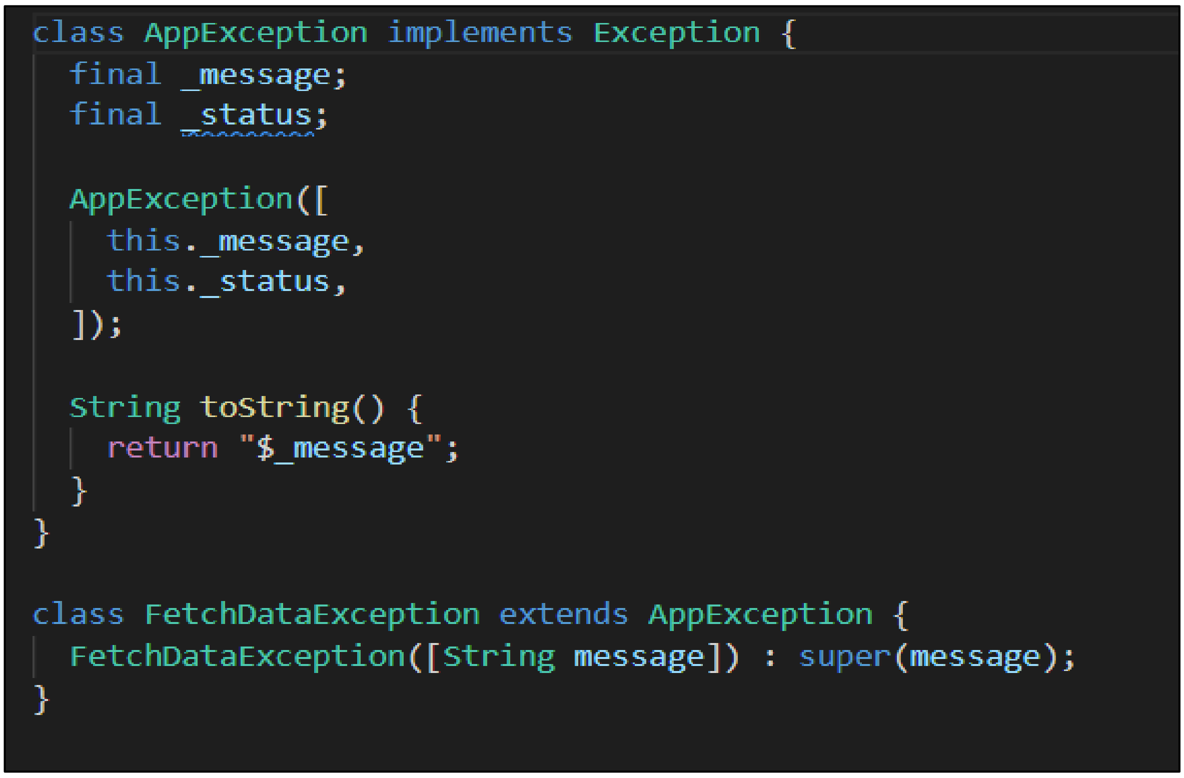
The snippet checks if the status code is "OK", if so then the json file is returned, else if the status code is "error" then the Fetch-Data Exception is thrown.



```
  getStatsResponse(StateLocation stateLocation,
      {String code = "", bool yesterday = false}) async {
    String endpoint = _getStatsEndpoint(
        location: stateLocation, code: code, yesterday: yesterday);
    String url = _apiService.statsUrl + endpoint;
    try {
      var response = await http.get(url);
      if (response.statusCode == 200) {
        // ignore: non_constant_identifier_names
        var Json = json.decode(response.body);
        if (stateLocation == StateLocation.TOP_FIVE) {
          return Json.sublist(0, 6);
        }
        return Json;
      } else {
        throw FetchDataException("Failed to load stats");
      }
    } on SocketException {
      throw FetchDataException("No internet connection");
```

Figure 7.6
App Client-2

The snippet is trying to find the specific information from the file received using endpoints. It checks if the status code is 200, if so then it decodes the json file, else it will throw FetchDataException.



```
class AppException implements Exception {
  final _message;
  final _status;

  AppException([
    this._message,
    this._status,
  ]);

  String toString() {
    return "$_message";
  }
}

class FetchDataException extends AppException {
  FetchDataException([String message]) : super(message);
}
```

Figure 7.7
Exceptions

The snippet shows how the exceptions that are thrown is handled.

8. RESULTS

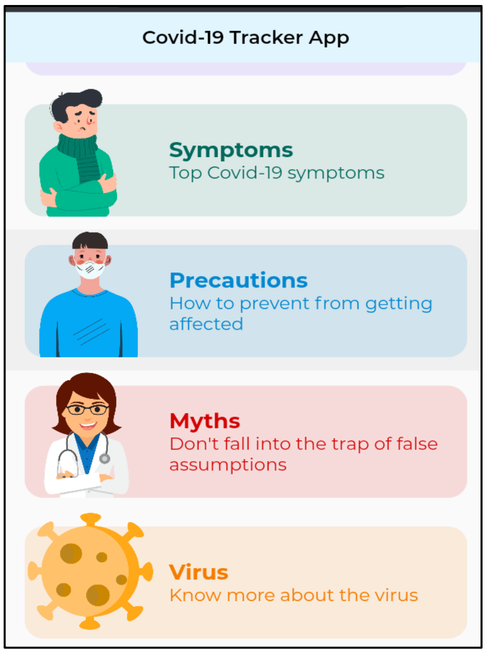

Covid-19 Tracker App
Symptoms
Top Covid-19 symptoms
Precautions
How to prevent from getting affected
Myths
Don't fall into the trap of false assumptions
Virus
Know more about the virus

Figure 8.1
Main Screen -1

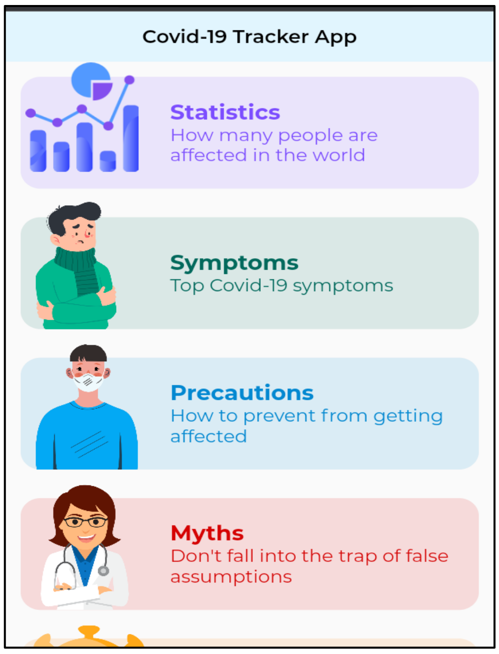

Covid-19 Tracker App
Statistics
How many people are affected in the world
Symptoms
Top Covid-19 symptoms
Precautions
How to prevent from getting affected
Myths
Don't fall into the trap of false assumptions

Figure 8.2
Main Screen -2

The two above screens represent the front screen containing the statistics, symptoms, precautions, Myths and Virus options.

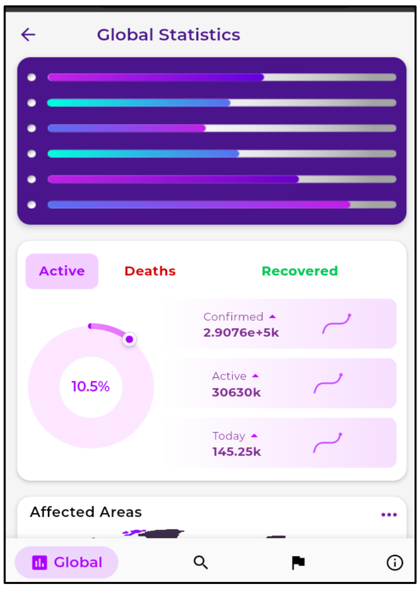


Figure 8.3
Statistics-Active

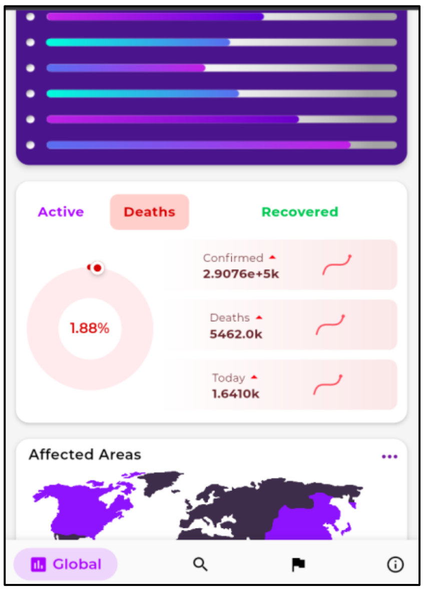
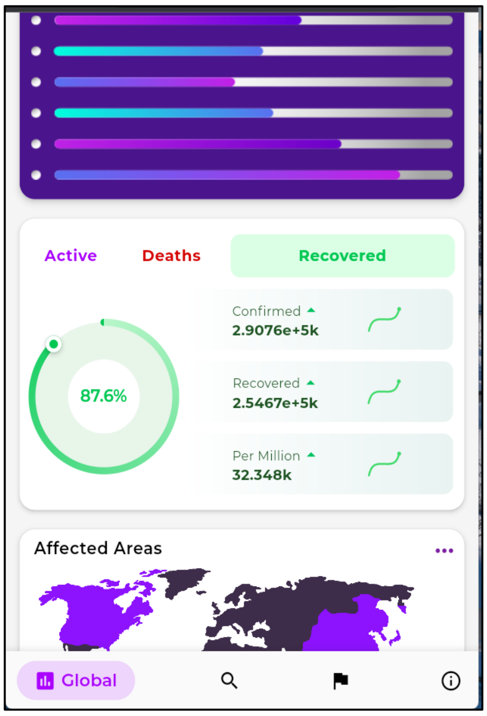


Figure 8.4
Statistics-Deaths

The above two screens represent the option statistics, where we can view the active and deaths statistics of the cases.



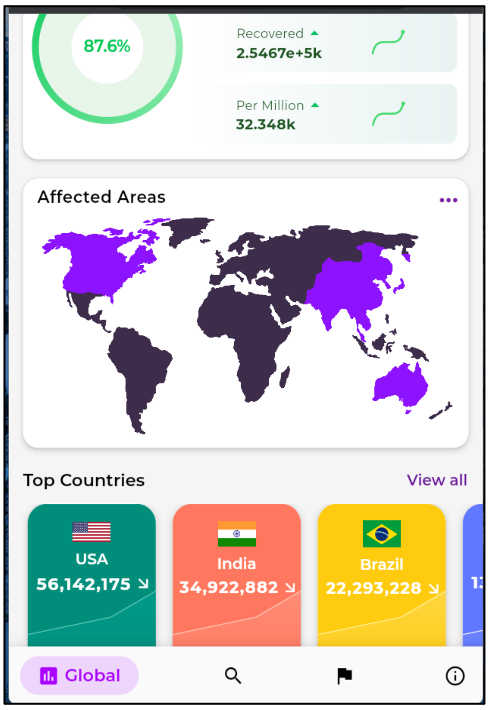
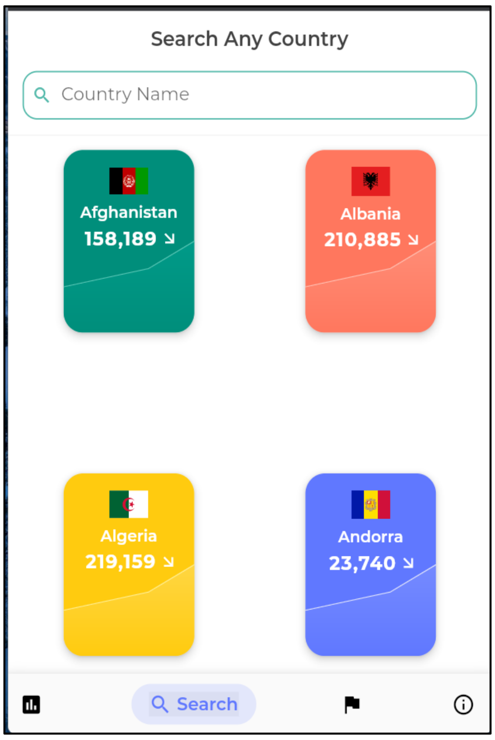


Figure 8.5
Statistics-Recovered

Figure 8.6
Statistics

The above two screens represent the options of statistics- the global recovered cases and the view countries option.



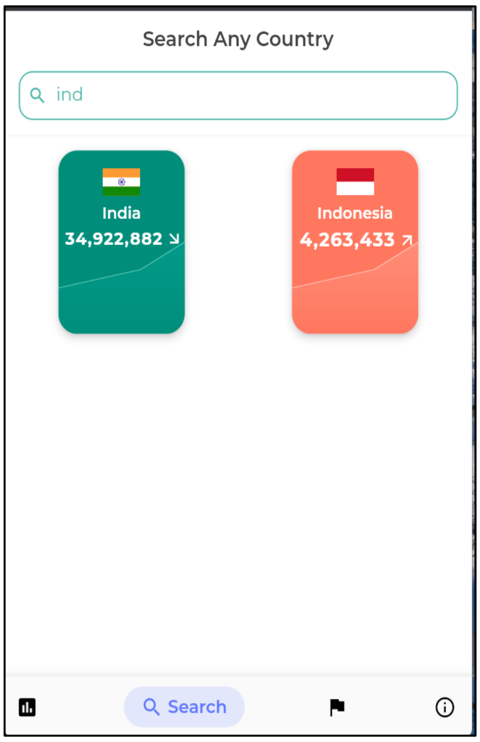
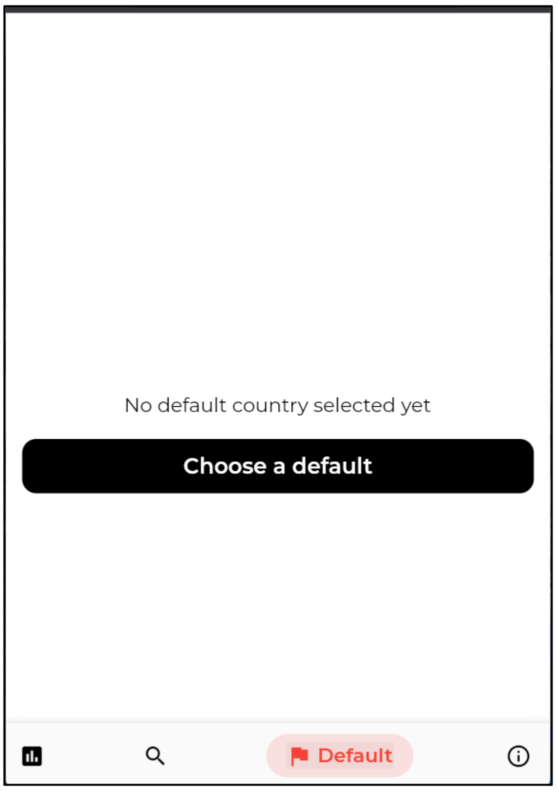
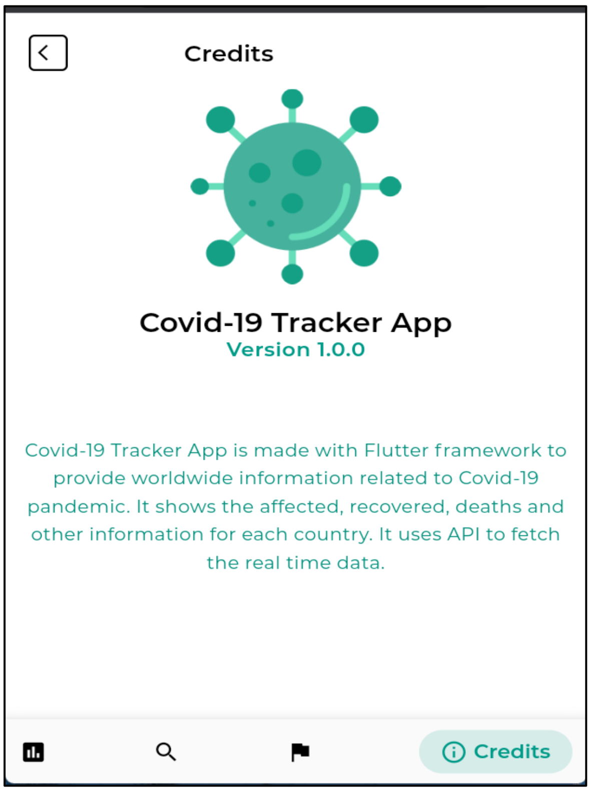


Figure 8.7
Statistics-Search Country-1

Figure 8.8
Statistics-Search Country-2

The above two screens represent the search tab of the statistics option allowing the user to search a particular country.



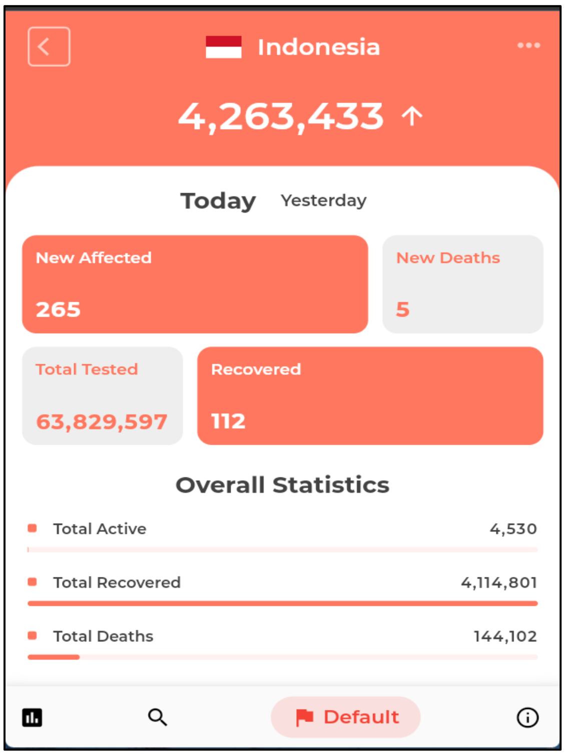


Covid-19 Tracker App is made with Flutter framework to provide worldwide information related to Covid-19 pandemic. It shows the affected, recovered, deaths and other information for each country. It uses API to fetch the real time data.

Figure 8.9
Statistics-default

Figure 8.10
Statistics- Credits

The two above screens represent the two tabs of statistics – making a country default makes it easy to track the cases and the credits of the app.



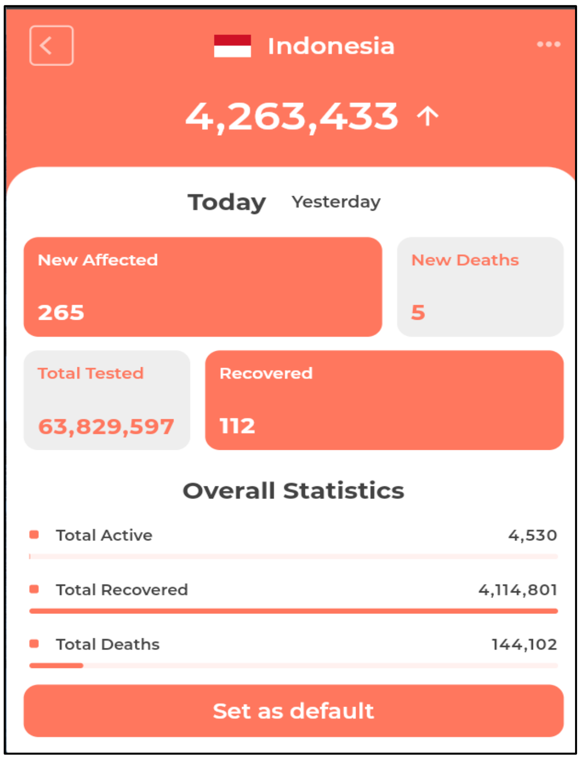
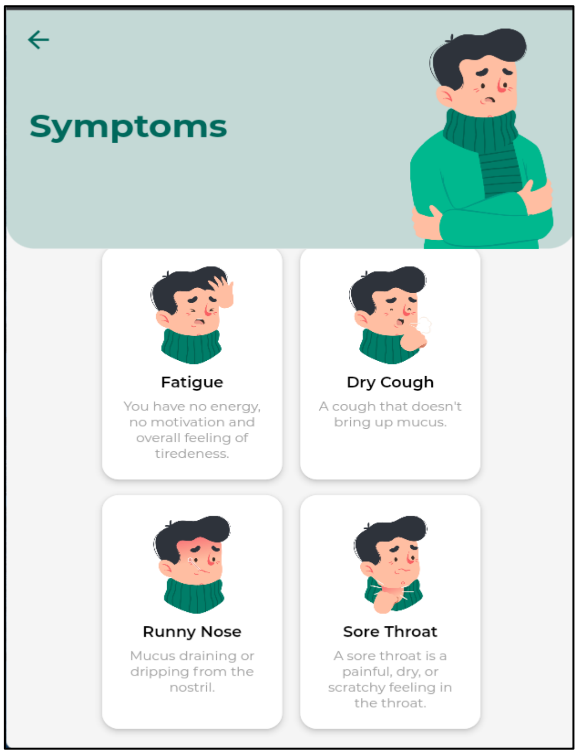


Figure 8.11
Country Statistics

Figure 8.12
Set a Country as default

The above two screens represent the process of making a country default.



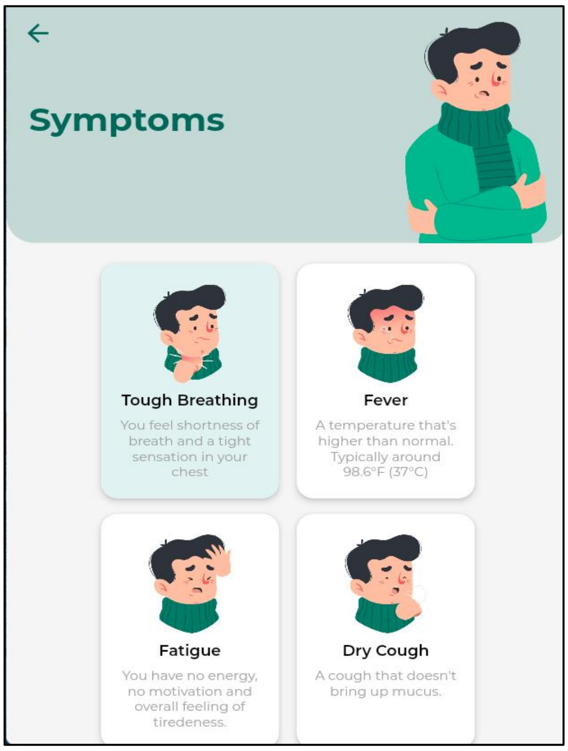
Symptoms

Fatigue
You have no energy, no motivation and overall feeling of tiredeness.

Dry Cough
A cough that doesn't bring up mucus.

Runny Nose
Mucus draining or dripping from the nostril.

Sore Throat
A sore throat is a painful, dry, or scratchy feeling in the throat.



Symptoms

Tough Breathing
You feel shortness of breath and a tight sensation in your chest

Fever
A temperature that's higher than normal. Typically around 98.6°F (37°C)

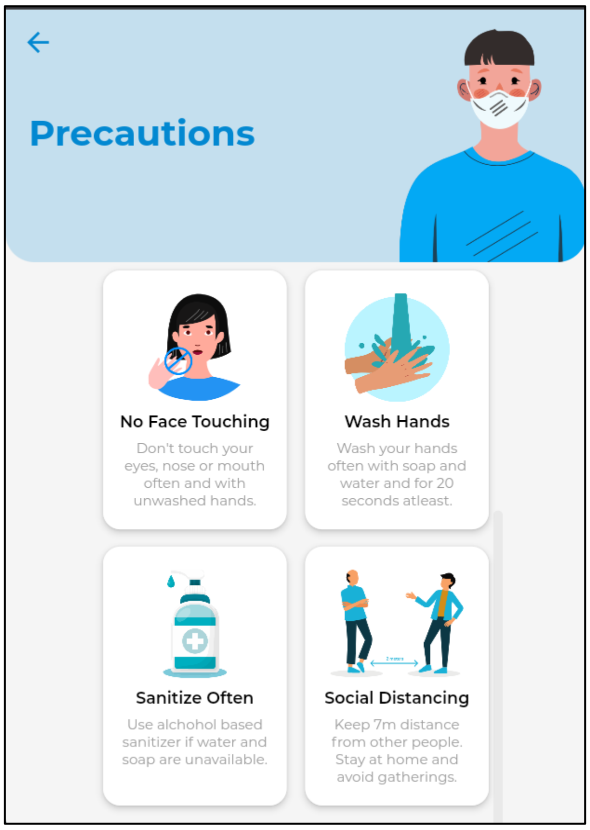
Fatigue
You have no energy, no motivation and overall feeling of tiredeness.

Dry Cough
A cough that doesn't bring up mucus.

Figure 8.13
Symptoms-1

Figure 8.14
Symptoms-2

The above two screens represent the symptoms option - details of symptoms of covid which will help the user analyse himself.



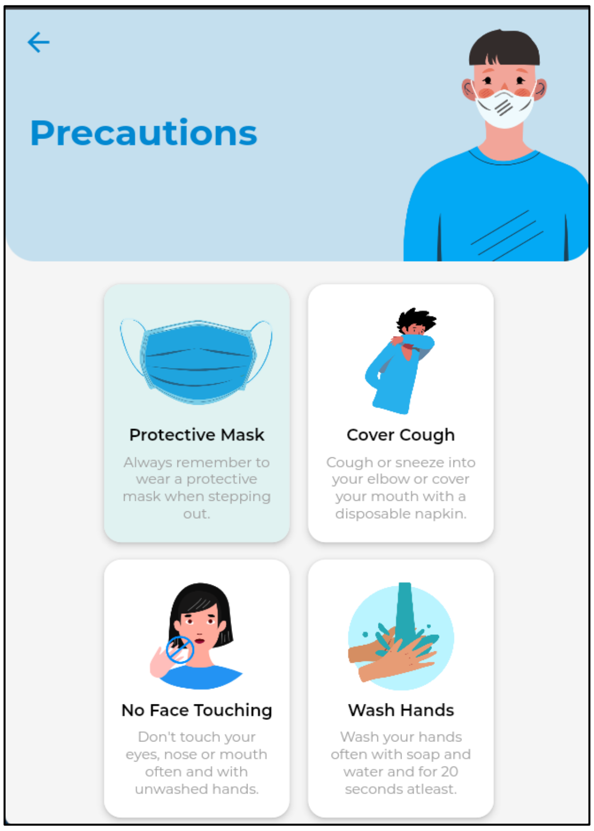
Precautions

No Face Touching
Don't touch your eyes, nose or mouth often and with unwashed hands.

Wash Hands
Wash your hands often with soap and water and for 20 seconds atleast.

Sanitize Often
Use alchohol based sanitizer if water and soap are unavailable.

Social Distancing
Keep 7m distance from other people. Stay at home and avoid gatherings.



Precautions

Protective Mask
Always remember to wear a protective mask when stepping out.

Cover Cough
Cough or sneeze into your elbow or cover your mouth with a disposable napkin.

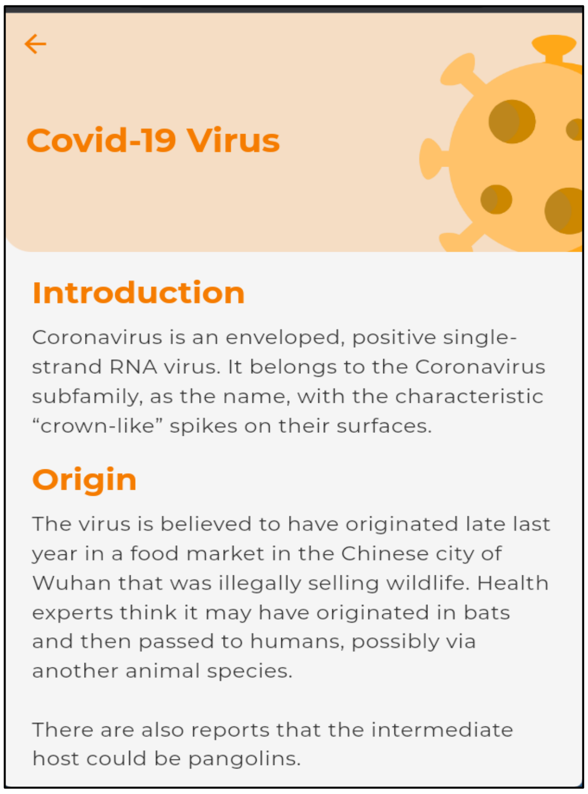
No Face Touching
Don't touch your eyes, nose or mouth often and with unwashed hands.

Wash Hands
Wash your hands often with soap and water and for 20 seconds atleast.

Figure 8.15
Precautions-1

Figure 8.16
Precautions-2

The above two screens represent the Precautions options - the various precautions that need to be taken care to avoid getting effected by covid.



Covid-19 Virus

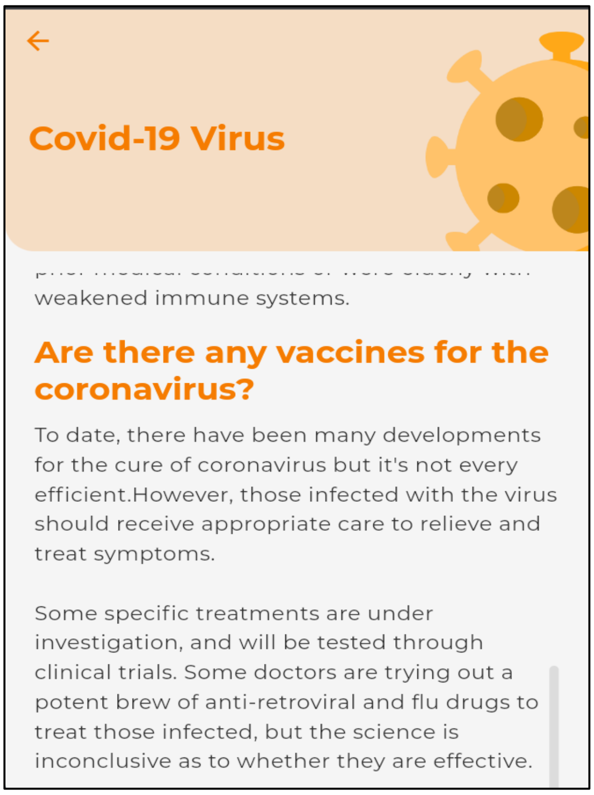
Introduction

Coronavirus is an enveloped, positive single-strand RNA virus. It belongs to the Coronavirus subfamily, as the name, with the characteristic "crown-like" spikes on their surfaces.

Origin

The virus is believed to have originated late last year in a food market in the Chinese city of Wuhan that was illegally selling wildlife. Health experts think it may have originated in bats and then passed to humans, possibly via another animal species.

There are also reports that the intermediate host could be pangolins.



Covid-19 Virus

weakened immune systems.

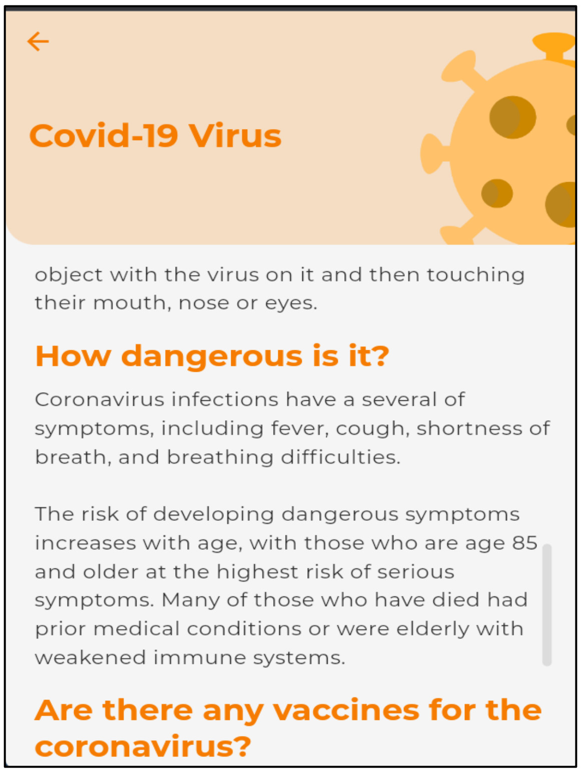
Are there any vaccines for the coronavirus?

To date, there have been many developments for the cure of coronavirus but it's not every efficient.However, those infected with the virus should receive appropriate care to relieve and treat symptoms.

Some specific treatments are under investigation, and will be tested through clinical trials. Some doctors are trying out a potent brew of anti-retroviral and flu drugs to treat those infected, but the science is inconclusive as to whether they are effective.

Figure 8.17
Virus -1

Figure 8.18
Virus -2



Covid-19 Virus

object with the virus on it and then touching their mouth, nose or eyes.

How dangerous is it?

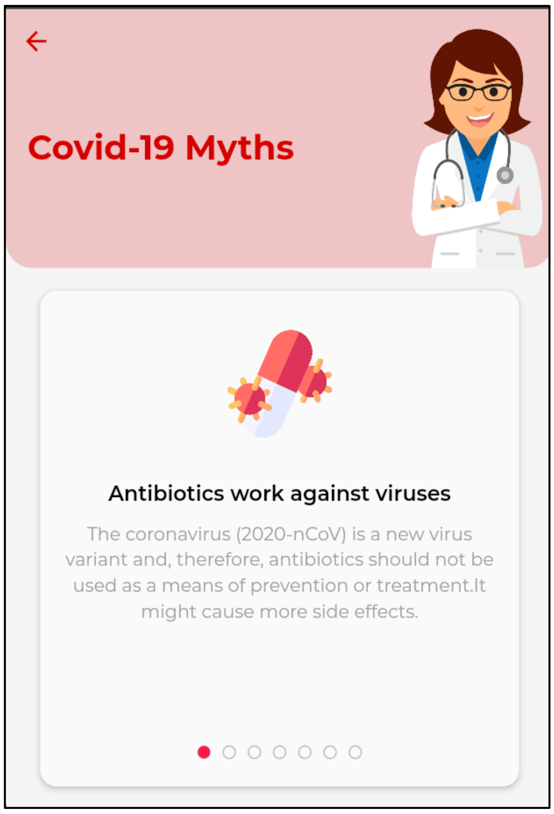
Coronavirus infections have a several of symptoms, including fever, cough, shortness of breath, and breathing difficulties.

The risk of developing dangerous symptoms increases with age, with those who are age 85 and older at the highest risk of serious symptoms. Many of those who have died had prior medical conditions or were elderly with weakened immune systems.

Are there any vaccines for the coronavirus?

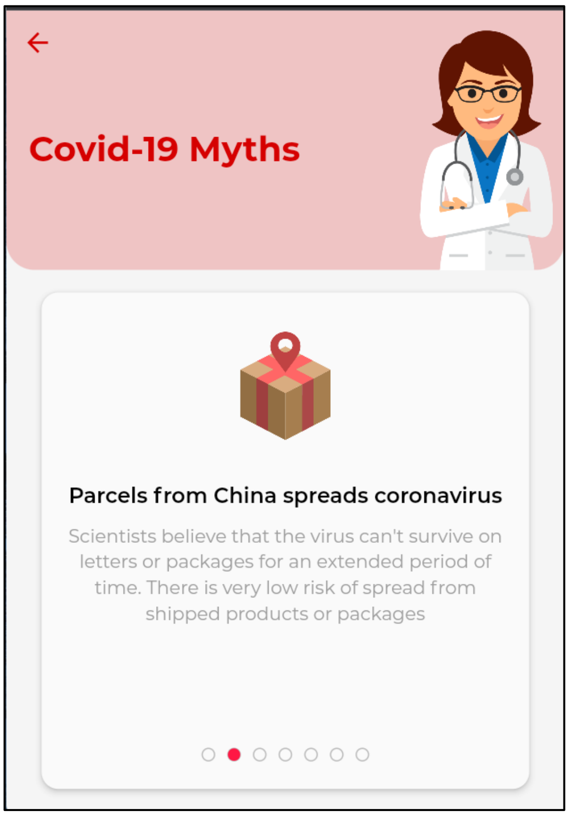
Figure 8.19
Virus-3

The above three screens represent the Virus Information option- which displays the information about the virus for the awareness of the public.


Covid-19 Myths

Antibiotics work against viruses

The coronavirus (2020-nCoV) is a new virus variant and, therefore, antibiotics should not be used as a means of prevention or treatment.It might cause more side effects.

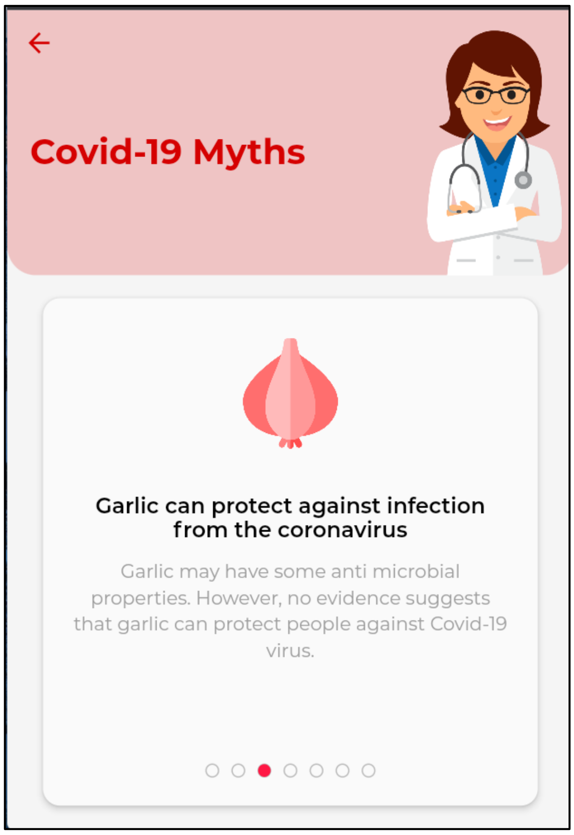

Covid-19 Myths

Parcels from China spreads coronavirus

Scientists believe that the virus can't survive on letters or packages for an extended period of time. There is very low risk of spread from shipped products or packages

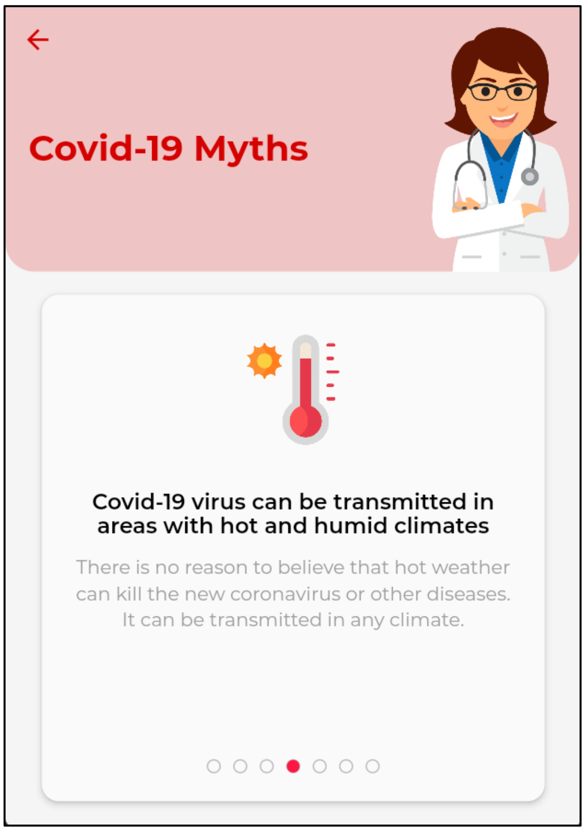
Figure 8.20
Myths-1

Figure 8.21
Myths-2


Covid-19 Myths

Garlic can protect against infection from the coronavirus

Garlic may have some anti microbial properties. However, no evidence suggests that garlic can protect people against Covid-19 virus.

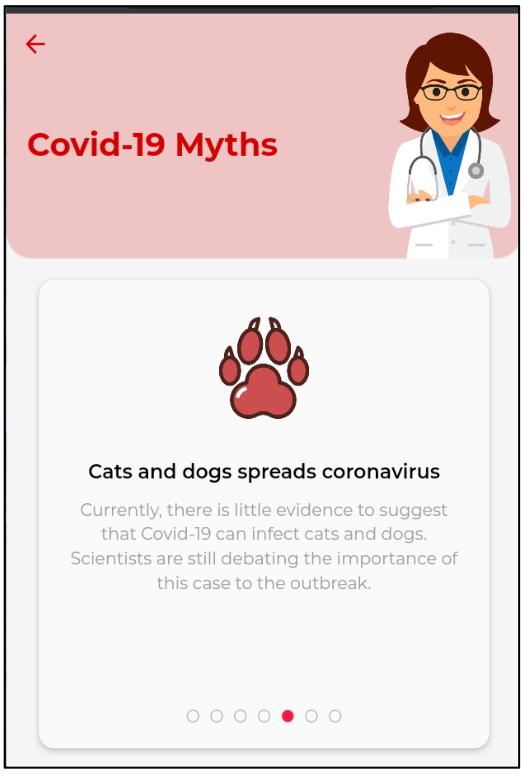

Covid-19 Myths

Covid-19 virus can be transmitted in areas with hot and humid climates

There is no reason to believe that hot weather can kill the new coronavirus or other diseases. It can be transmitted in any climate.

Figure 8.22
Myths-3

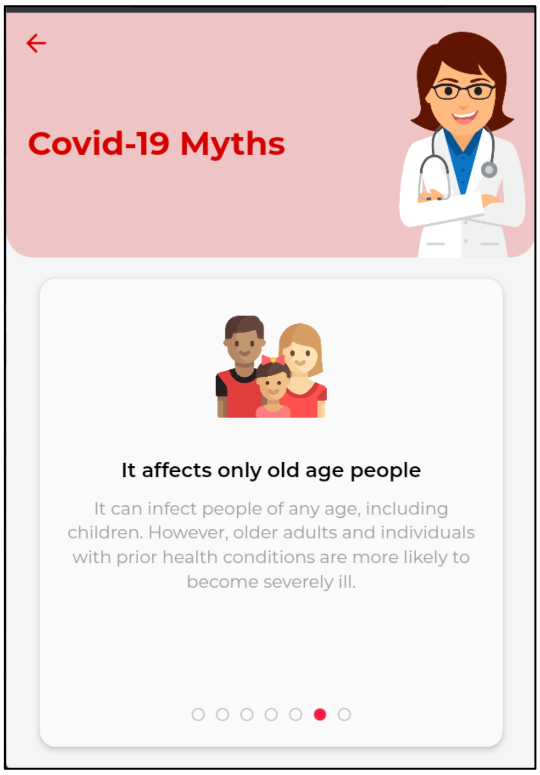
Figure 8.23
Myths-4


Covid-19 Myths

Cats and dogs spreads coronavirus

Currently, there is little evidence to suggest that Covid-19 can infect cats and dogs. Scientists are still debating the importance of this case to the outbreak.

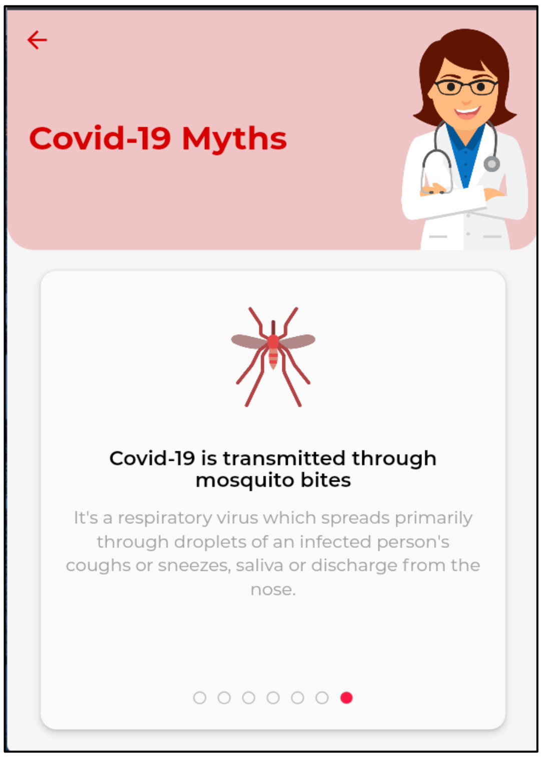
Figure 8.24
Myths-5


Covid-19 Myths

It affects only old age people

It can infect people of any age, including children. However, older adults and individuals with prior health conditions are more likely to become severely ill.

Figure 8.25
Myths-6


Covid-19 Myths

Covid-19 is transmitted through mosquito bites

It's a respiratory virus which spreads primarily through droplets of an infected person's coughs or sneezes, saliva or discharge from the nose.

Figure 8.26
Myths-7

The above 8.20- 8.26 screens represent the Myths options- which mentions the various misconceptions that exist about the virus which is required to be cleared for the betterment of the public.

9. CONCLUSION AND FUTURE WORK

9.1 Conclusion

The app provides efficient means to find information related to COVID-19 by displaying most necessary details to the user and is beginner friendly. It successfully provides the digital protection of the society, creates public awareness, and helps users to be up to date with essential details.

9.2 Future Work

The app can be extended to-

- Use APIs to provide covid-19 news on different countries and default country
- Make the app accurate
- Provide geo-locations
- Make it contact covid-19 centers
- Allow the users to book vaccination slots
- Take basic self-assessment tests by users
- Provide information on nearby containment zones

10. REFERENCES

- https://flutter.dev/learn
- https://docs.flutter.dev/reference/tutorials
- https://www.tutorialspoint.com/flutter/index.htm
- https://www.geeksforgeeks.org/flutter-material-design/
- https://www.ibm.com/cloud/learn/rest-apis
- https://www.raywenderlich.com/24499516-getting-started-with-flutter
- https://www.w3adda.com/flutter-tutorial
- https://fluttertutorial.in/

Reference IEEE Papers

- Velickovic, Z. S., Velickovic, M. Z., & Milivojevic, Z. N. (2021). Application of a Reliable Web API's in the Fight Against COVID-19 Infodemia. 2021 25th International Conference on Information Technology (IT).
- Mohammad Monirujjaman Khan, Rezaul Karim(2020). Development of Smart e-Health System for Covid-19 Pandemic. 2020 23rd International Conference on Computer and Information Technology (ICCIT).
- Keshav Kulsresth; Shivam Shasheesh; Chandan Mishra; K P Arjun. Covid 19 Tracker Using REST API Android App. 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT).
- Sukanya Chatterjee School of Computer Science and Engineering Lovely Professional University, Punjab, India; Mir Mohammad Yousuf; Manan Rasool; Ankit Chaurasiya; Mohammad Faisal; Vivek Pandey. Covid-19 Outbreak Cases and Healthcare Database. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM).
- Shady Boukhary; Eduardo Colmenares. A Clean Approach to Flutter Development through the Flutter Clean Architecture Package. 2019 International Conference on Computational Science and Computational Intelligence (CSCI).