

COVID19 Tracker System

AROMAL JOSEPH A
ARAVINDH M
LALITH KISHORE VP
KESHAVARDHAN DR
MOHAN RAJ MA

COVID-19 Tracker App Using REST API and Volley (Global Stats Only)



1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the COVID19 tracker system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

1.2. Scope of Project

☒ This application should have a provision to maintain a database for individual information, public information and COVID19 portfolio.

☒ Also, an integrated platform required for government and public.

☒ Filters like Low to High or showcasing COVID19 affected area should be highlighted.

☒ Aadhar integration for intimating individual reports to the public.

☒ Clarified that all contact tracing and COVID-19 status apps, including ones that merely store an individual's vaccination or test records, need to complete the "COVID-19 contact tracing and status apps" section in the App content page.

☒ Additional requirements are introduced for apps that plan to remove contact tracing or COVID-19 status functionalities.

☒ Apps that provide medical, treatment, vaccine, testing, or other related information for COVID-19.

☒ Apps that support COVID-19-related response, containment, research, or education/training efforts.

☒ Apps that support services used to respond specifically to COVID-19, for example, apps that provide social support (food stamps, payment), healthcare, loans, etc., specifically in response to COVID-19.

ABOUT THE APP

Contact Tracing

The app records if users are in close contact with another app user. If an app user tests positive for COVID-19 the app will alert other app users that have been closer than 2 metres for more than 15 minutes. If users choose to share their phone number, the HSE can phone to tell them what they need to do to keep themselves and others safe.

Symptom Tracking:

Daily Health Check-in Users can help slow the spread of COVID-19 by telling the HSE how they are feeling every day. Anonymous information about how many people have symptoms, at any time, helps us map and predict the spread of the virus. If users do have symptoms, then the app will give them advice on what to do.

Definitive Source of News & Information

The app will also give users easy access to the latest facts and figures about COVID-19 in Ireland and signpost them to information that will help them care for themselves if they are sick.

How this app is formally notified of a positive test diagnosis.

The app downloads the anonymous IDs that are shared with the HSE by people who have tested positive for COVID-19 every 2 hours. If these anonymous IDs match any of the anonymous IDs that are recorded on a user's phone they are alerted that they are a close contact.

The criteria for a close contact with someone who tests positive.

The app uses the European Centre for Disease Control's definition for a close contact, which includes persons within 2 metres of a confirmed case of COVID-19 for more than 15 minutes.

How users are contacted if they are in close contact with someone who tests positive.

The app sends an in-app notification when the app identifies that a user has been in close contact with a confirmed case. If they choose to share their phone number with the HSE the Contact Tracing Service can contact them by phone.

Information that users get if they have been in close contact with someone who tests positive.

If the app identifies that an app user is a close-contact they will get advice about restricting their movements to help stop the spread of the virus.

STRUCTURE OF THE APP

Key points to remember:

1. The id (for frontend) and attributes(backend) mentioned in the SRS should not be modified at any cost. Failing to do may fail test cases.
2. Remember to check the screenshots provided with the SRS. Strictly adhere to id mapping and attribute mapping. Failing to do may fail test cases.
3. Strictly adhere to the proper project scaffolding (Folder structure), coding conventions, method definitions and return types.

Adhere strictly to the endpoints given below.

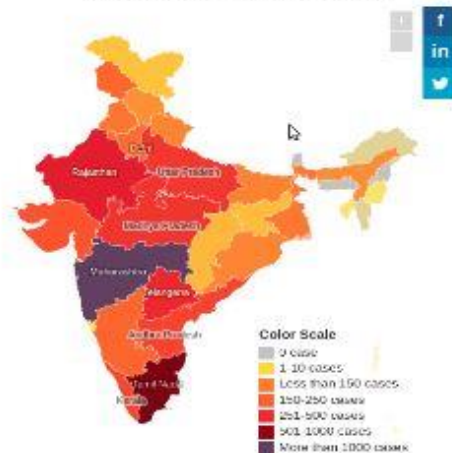
Application assumptions:

1. The login page should be the first page rendered when the application loads.
2. Manual routing should be restricted by using AuthGuard by implementing the canActivate interface. For example, if the user enters as `http://localhost:8000/signup` or `http://localhost:8000/home` the page should not navigate to the corresponding page instead it should redirect to the login page.
3. Unless logged into the system, the user cannot navigate to any other pages.
4. Logging out must again redirect to the login page.
5. To navigate to the admin side, you can store a user type as admin in the database with a username and password as admin.
6. Use admin/admin as the username and password to navigate to the admin dashboard.

Data Visualization of Covid-19 cases in India (State and District wise) by IIT Bombay Nutrition Group

State-wise confirmed COVID-19 cases in India

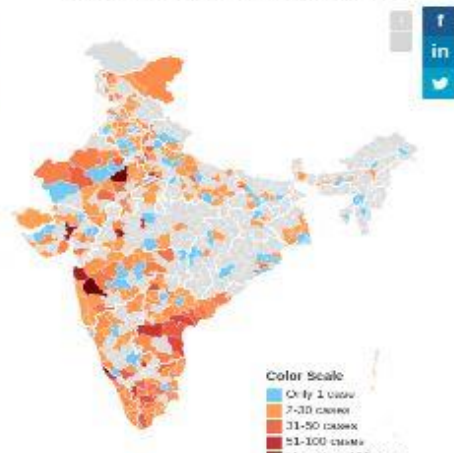
Put your mouse pointer on states for details



(Updated on 11 April 2020, 05:09 PM)
Source: Ministry of Health and Family Welfare - MoHFW
- Created with Datawrapper

District-wise confirmed COVID-19 cases in India

Put your mouse pointer on districts for details



As per the Latest Update on 11th April 2020 from MoHFW website
Source: Ministry of Health and Family Welfare - MoHFW
- Created with Datawrapper

FRONTEND

Customer:

1. Auth: Design an auth component (Name the component as auth for angular app whereas Auth for react app. Once the component is created in react app, name the jsx file as same as component name i.e Auth.jsx file) where the customer can authenticate login and signup credentials
2. Signup: Design a signup page component (Name the component as signup for angular app whereas Signup for react app. Once the component is created in react app, name the jsx file as same as component name i.e Signup.jsx file) where the new customer has options to sign up by providing their basic details.

a. Ids:

☐ Aadhar number

- ☐ mobilenumber
- ☐ password
- ☐ confirmpassword
- ☐ submitButton
- ☐ signinLink
- ☐ signupBox

b. API endpoint Url: <http://localhost:8000/signup>

c. Output screenshot:

3. Login: Design a login page component named (Name the component as login for angular app whereas Login for react app. Once the component is created in react app, name the jsx file as same as component name i.e Login.jsx file)where the existing customer can log in using the registered email id and password.

a. Ids:

- ☐ Aadhar number
- ☐ password
- ☐ submitButton
- ☐ signupLink
- ☐ loginBox

b. API endpoint Url: <http://localhost:8000/login>

c. Output screenshot:



4. Dashboard / Home: Design a home page component named (Name the

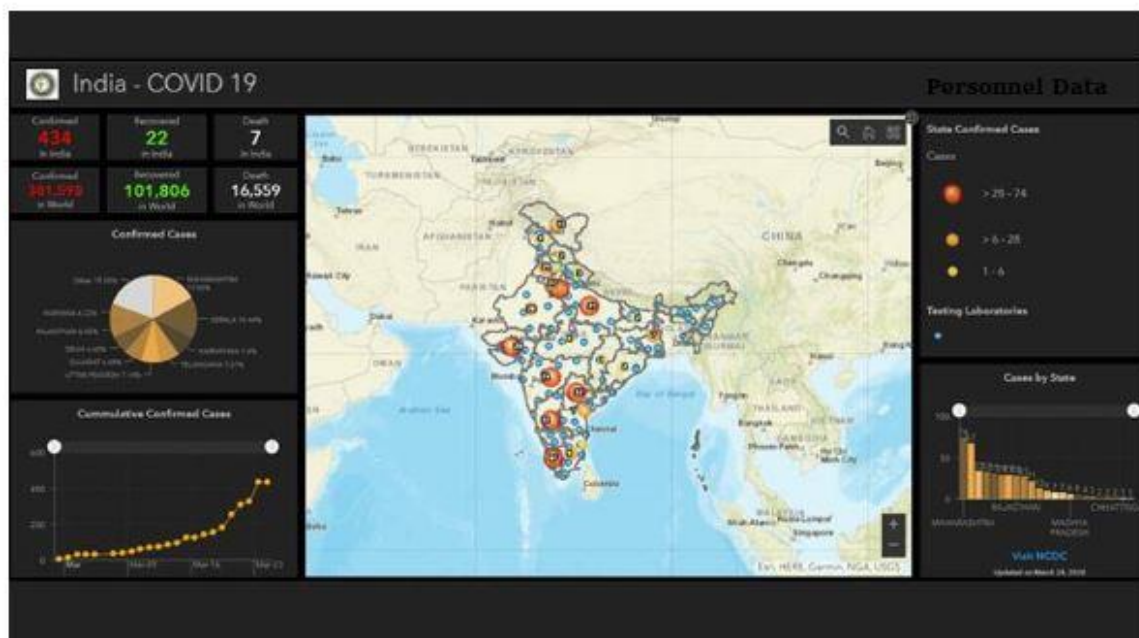
component as homepage for angular app whereas HomePage for react app. Once the component is created in react app, name the jsx file as same as component name i.e HomePage.jsx file) that has the navigation bar

a. Ids:

1. userNavbar
2. HomeButton
3. Personnel data
4. Over all State data
5. logoutButton

b. API endpoint Url: <http://localhost:8000/home>

c. Screenshot



Admin:

6. Admin Dashboard: Design a dashboard page named (Number of affected as dashboard for angular app whereas Dashboard for react app. Once the numbers

created in react app, name the jsx file as same as component name i.e

Dashboard.jsx file) where the number of affected persons is displayed on the admin side.

a. Admin Navigation: Design a navigation component (Name the component as adminhomepage for angular app whereas AdminHomePage for react app.

i.Ids:

1. adminNavbar
2. adminaddtButton
3. adminconfirmButton
4. logoutButton

b. Add number of patients affected: Design an add product component (Name the component as addpatient for angular app whereas AddPatient for react app.

1.addnumber of affected count

2.StateName

3.District Name

4Aadhar Number

5.affected

6.Recovered

7.adddataButton

ii.API endpoint Url: <http://localhost:8000/addProduct>



Backend:

Class and Method description:

Model Layer:

1. **UserModel**: This class stores the user type (admin or the customer) and all user information.

a. Attributes:

- i. Aadhar: String
- ii. password: String
- iii. mobileNumber: String
- iv. active: Boolean
- v. role: String

2. **LoginModel**: This class contains the email and password of the user.

a. Attributes:

- i. Aadhar: String
- ii. password: String

3. **Covid 19 Model:** This class stores the details of the patient.

a. **Attributes:**

- i. StatelId: String
- ii. imageUrl: String
- iii. patient name: String
- iv. Status: String

Controller Layer:

6. **SignupController:** This class control the user signup

a. Methods:

i. saveUser(UserModel user): This method helps to store users in the database and return true or false based on the database transaction.

7. **LoginController:** This class controls the user login.

a. Methods:

i. checkUser(LoginModel data): This method helps the user to sign up for the application and must return true or false.

8. **Patient Controller:** This class controls the add/edit/update/view number of person affected by Covid 19.

a. Methods:

i. List<state>; getState(): This method helps the admin to fetch all datas from the database.

ii. List<District>; getDistric (): This method helps to retrieve all the datas from the database.

iii. Patient Details EditData(String id): This method helps to retrieve a affected details from the database based on the Aadhar id.

iv. Patient Details Edit Save(Patient Details data): This method helps to edit

a Patient Details and save it to the database.

v. Patient Details Save(Patient Details data): This method helps to add a new Patient Details to the database.

vi. Patient Details Delete (String id): This method helps to delete a Patient Details from the database.