SOFTWARE REQUIREMENT SPECIFICATION

FOR

COVID-19 TRACKER

***OBJECTIVE:***

* Building an application for covid-19 open API tracker site

***USERS OF THE SYSTEM:***

1. Government
2. Public

***FUNCTIONAL REQUIREMENTS:***

* Build an application by which public can examine about covid-19 cases.
* Application should have a sign up, login, profile, dashboard page and individual records.

1. State name
2. Confirmed cases
3. Recovered cases
4. Death
5. Fatality rate
6. Latitude
7. Longitude
8. Last updated

* Online authentication to verify the identity claim of aadhar holder.
* Application should have facilities to store database of individual and covid-19 portfolio.
* Application should provide informations about medical, treatement, vaccination and all other covid-19 informations

While the above ones are basic functional features expected, the below ones can be nice to have add-features,

* Application should give pictorial representation of the comparative study about covid-19 status(confirmed,death,recovered) at weekends.
* Application should be accessible by both androids and windows.
* The affected individuals can fetch information about the bed availability in near by hospitals and treat themselves.
* Application should have facility to get comments from the public.

***OUTPUT/POST CONDITION:***

* Standalone application
* Record persisted in success & failure collections

***NON-FUNCTIONAL REQUIREMENTS:***

* All mandatory verifications should be done in order to access the application.
* The system need permenant internet connectivity.

**SECURITY:**

* Apps created specifically for covid-19 response may not access personal and sensitive data that’s not required.
* Apps must have publicly accessible privacy policy.
* All the informations of the public should be secured.

***STANDARD FEATURES:***

* Number of tested
* Confirmed cases
* Deaths in the country
* A heatmap of the largest concentrations of confirmed covid-19 cases.
* Location of the public testing centers in each state.
* Latest news updates.
* Operating from any time at any location using wifi or cellular network.

***LOGGING AND AUDITING:***

The system should support logging(app or web) and auditing at all levels.

***MONITORING:***

Should be monitorised via enterprise monitoring tools.

***CLOUD:***

The solution should be made cloud ready & solution have a minimum impact when moving away to cloud infrastructure.

***BROWSER COMPATIBLE:***

All latest browsers.

***TECHNOLOGY STACK:***

* HTML&CSS
* Javascript

***KEYPOINTS TO REMEMBER:***

* The id(for frontend) and attributes(backend) mentioned in the SRS should not be modifies at any cost. Failing to do may fail test cases.
* Remember to check the screenshots provided with the SRS. Strictly adhere to id mapping and attribute mapping. Failing to do may fail test cases.
* 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:***

* The login page should be the first page to be opened when the application loads.
* Unless logged into the system, the user cannot navigate to any other pages.
* Logging out must again redirect to login page.
* Use admin/admin as the username and password to navigate to the admin dashboard.

***VALIDATIONS:***

* Basic aadhar validation should be performed.
* Basic email validation should be performed.
* Basic mobile validation should be performed.

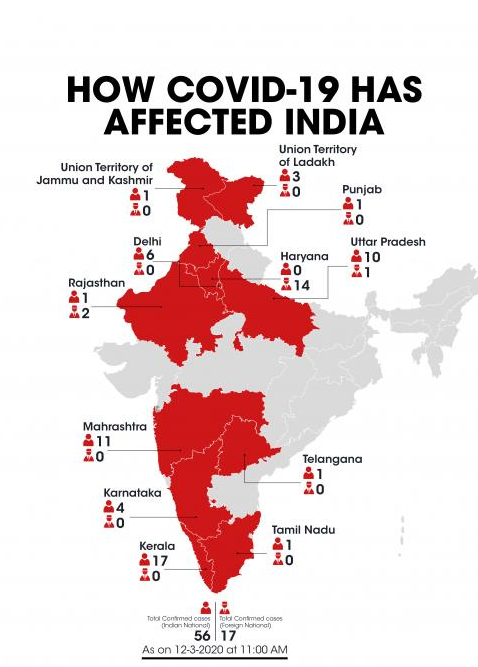
***PROJECT TASKS:***

* Complete the “COVID-19 contact tracing and status apps” sections in app content page
* Submit proof of eligibility via the advance notice form
* Privacy requirements
* App visibility and user awareness
  + For apps that collect information in the foreground or use foreground service
  + For apps that collect information when running as a background service.
* API requirements
* Editorial and quality requirements
* App review and visibility

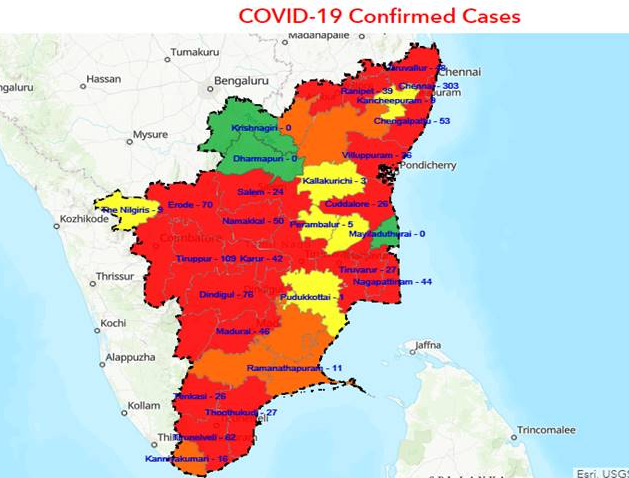
***USER:***

COVID-19 HEAT MAP

**STATE WISE CONFIRMED COVID CASES:**

******

**DISTRICT WISE CONFIRMED COVID CASES:**

******

***FRONT END:***

***USER:***

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 user has options to sign up by providing their basic details.

a***.   Ids:***

* + Aadhar number
  + Mobile number
  + password
  + confirm password
  + submit Button
  + signin Link
  + signup Box

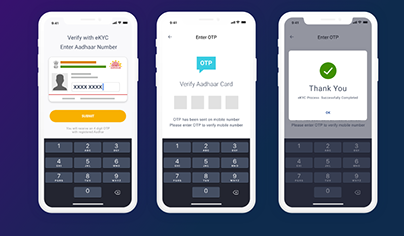
b***.   API endpoint Url***: http://localhost:8000/signup

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



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

component as homepagefor 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. user Navbar

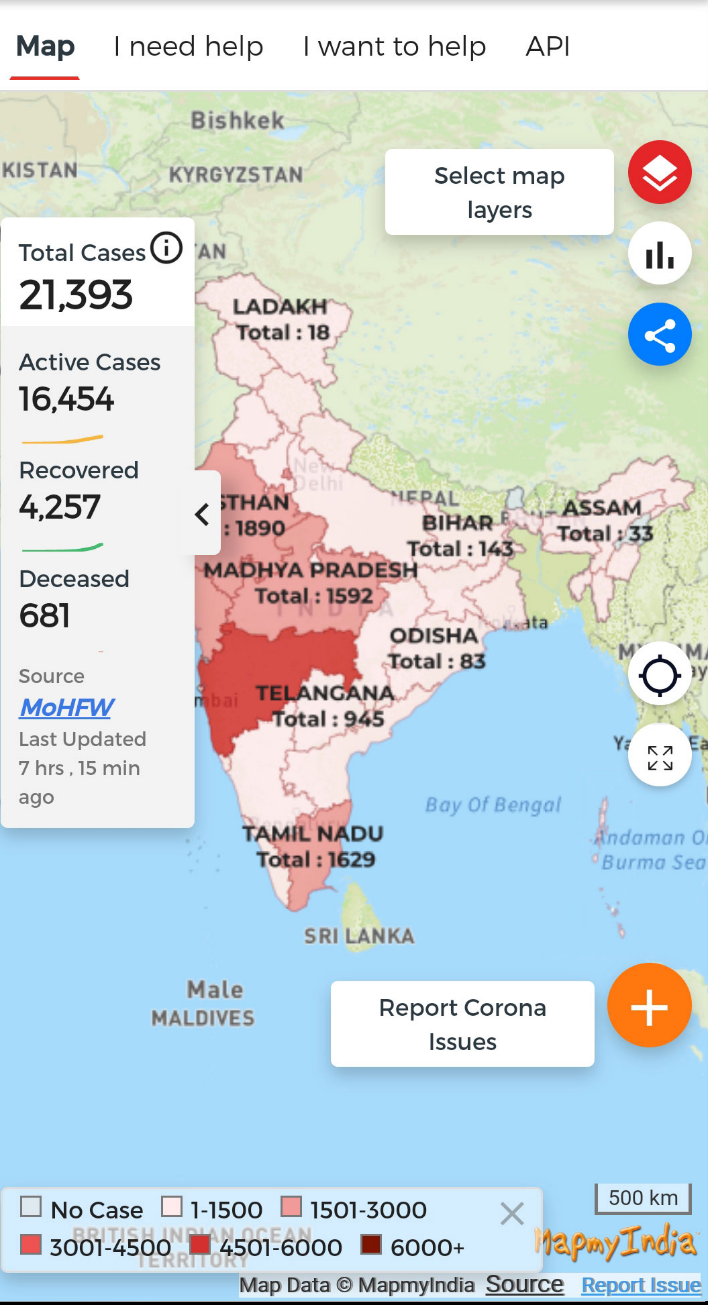
2. Home Button

3. Personnel data

4. Over all State data

5. logout Button

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



***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. admin Navbar

2. admin addt Button

3. admin confirm Button

4. logout Button

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. State Name

3. District Name

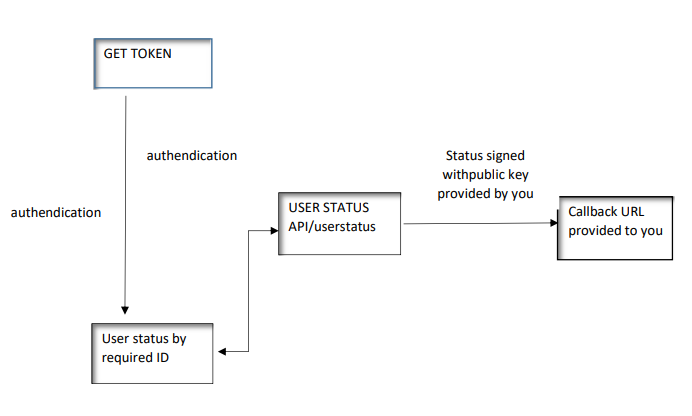
4. Aadhar Number

5. Affected

6 .Recovered

7. Add data Button

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.   StateId: 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&lt;state&gt; getstate():** This method helps the admin to fetch all datas from the database.

                ii.   **List&lt;District&gt; 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.