**SRS for application to track Covid-19 density in India**

**Objective:**

Building an application for covid 19 open API tracker site

**Users of the application:**

1. App developed community which represents the government

2.public people

**Functional Requirements:**

* Build an application through which public can access and check the covid 19 status
* The application should have login, sign up, profile, dashboard for each and every individual.
* It should contain,
* Individual's state name
* Number of deaths in their state
* Number of confirmed cases in their state
* Fatality rate
* Latitude
* Longitude
* Latest update
* It should have to maintain a database which contains information of each person, public information and covid 19 portfolio
* It should give alert to the user when he/she is near the covid hotspot areas
* It should also be an integrated platform for government and public
* It should give the covid status and statistics as notification for each user every 24 hours
* It should also show a demo video for users on how to maintain hygiene and safety to avoid covid
* It should intimate safety precautions every 24 hours for users
* It should show the health rate of the users and alert if they are in the verge of covid affecting

**Output/post condition:**

* Records Persisted in Success & Failure Collections
* Standalone application / Deployed in an app Container

**Non-Functional Requirements:**

**Security**

* App must have a safe digital locker to secure the information of users
* It should maintain the affected persons personal information in the way others can’t see it.

**Standard Features**

* Number of tested
* Confirmed cases
* Deaths in the country
* A heatmap of the largest concentrations of confirmed covid-19 cases
* Locations of public testing centers in each state
* News updates from major health org’s
* Alert notification for every 24 hours
* Tracking users location using gps and alerting them

**Logging**

* The system should support logging(app/web/DB)at all levels

**Cloud**

* The solution should be made cloud-ready and should have a minimum impact when moving away to cloud infrastructure

**Browser compatible**

All latest browsers with advanced accessibilities'

**Technology stack**

* HTML&CSS
* JavaScript

**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 demo video and safety precautions should be at the first page followed by login page

2. Manual routing should be restricted by using Auth Guard by implementing

the can Activate interface. For example, if the user enters

as [https://fightwithcovid/signup](https://manvscovid/signup) or https://fightwithcovid/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

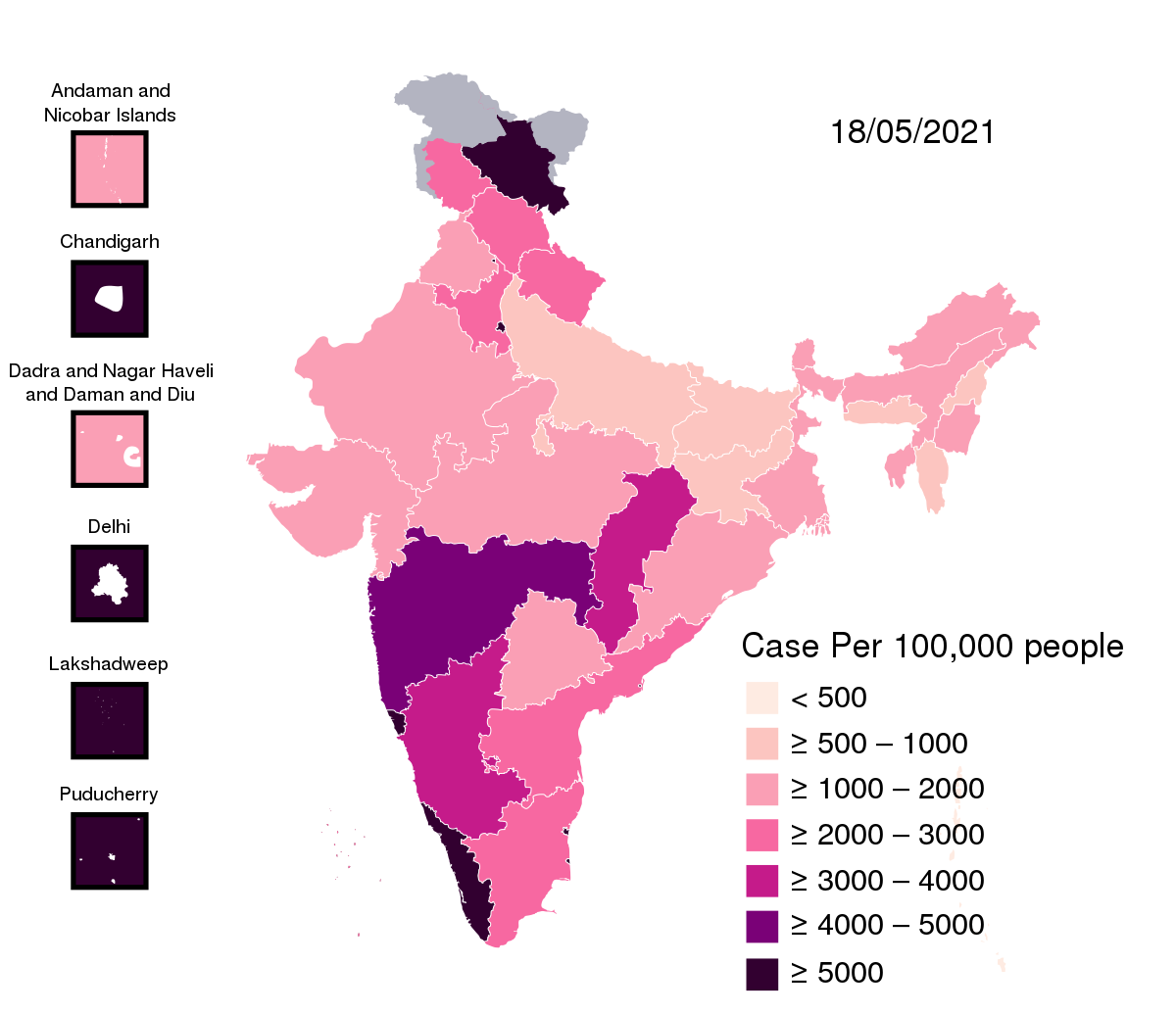
database with a username and password as admin.

6. Use admin/admin as the username and password to navigate to the admin

Dashboard.

**Validations:**

Basic mobile validation should be performed

Aadhaar validation should be performed

**Frontend:**

**Customer:**

1. S**ignup:** design a signup page component(name the component as signup foe angular app and 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

* User name
* Aadhaar number
* Mobile number
* Password
* Confirm password
* Submit button
* Sign in link
* Signup box

**b. API endpoint URL:https://fightwithcovid/signup**

C. **Output screensho**t

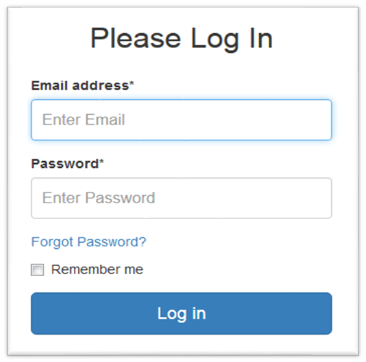
2.**Login:** Design a login page component named (Name the component as login for angular app whereas Login for react app. Once the component is created in react app, name the jsx file as same as component name I.e Login.jsx file) where the existing customer can log in using the registered username and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

a. IDs

* Username
* Password
* Submit button
* Signup link
* Login box

**b. API endpoint URL**: <https://fightwithcovid/login>

**c. output screenshot**



**3.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**

* User Navbar
* Home Button
* Personal data
* Their state data
* Overall state data
* Logout button

**b.API endpoint URL:** <https://fightwithcovid/home>

**c.output Screenshot**

**Admin:**

**5. 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 and number of persons using their app are affected are 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:**

⦁ admin Navbar

⦁ admin add Button

⦁ admin confirm Button

⦁ logout Button

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

1.add number of affected count

2.State Name

3.District Name

4.Aadhar Number

5.affected

6.Recovered

7.deaths

8.add data Button

**ii.API endpoint Url:** http://fightwithcovid/addProduct

**c.Screenshot**

**Backend:**

**Class and Method description:**

**Model Layer**:

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

a. Attributes:

1.Username:String

2. Aadhar: String

3.password: String

4. mobile Number: String

5. active: Boolean

6. role: String

2.  **Login Model**: This class contains the username and password of the user.

a. Attributes:

1. Username: String

2.password: String

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

a. Attributes:

1. State Id: String

2 image Url: String

3 patient name: String

4 Status: String

**Controller Layer:**

6. **Signup Controller**: This class control the user signup

a. Methods:

1. save User(User Model user): This method helps to store users in the database and return true or false based on the database transaction.

7.  **Login Controller:** This class controls the user login.

a. Methods:

1. **check User(Login Model 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> get state(): This method helps the admin to fetch all data's from the database.

ii. List<District> get District(): This method helps to retrieve all the data's from the database.

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

vii. Patient alert: This method helps to alert patient to be isolated