Covid -19 cases across india

Objective :

Building an UI for covid -19 cases tracker page

Users of the system :

1. Government
2. Public

Functional Requirements:

* Building an application that both public and government can access and view the covid -19 status in india.
* The application consist of signup , login ,profile ,dashboard page and individual records.

1. State name.
2. Confirmed cases.
3. Death cases.
4. Fatality cases.
5. Latitude.
6. Longitude.
7. Last updates.

* This application have a provision to maintain a database for individual information , public information and covid -19 portfolio.
* It is also an integrated platform required for government and public.
* It also contain filters likelow to high or showcasing covid -19 affected area should be highlighted.
* Aadhar integration for intimating individual reports to the public.
* Clarified that all contact tracing and covid-19status app, including ones that merely store an individual’s vaccination or test records , ned to complete the covid -19 contact tracing and status apps
* Additional requirements are introduced for apps that plan to remove contact tracing or covid -19 status functionalities.
* This application will provide medical, treatment , vaccine ,testing , other related information for covid-19.
* This app will support covid-19 realated response , containment , research or education/training efforts.
* This app will support sevices used to respond specifically to covid-19.
* For example this app will provide social support like (food stamps , payment, health care, loans ..etc).

Output / post condition :

* This records persisted in success & failure collections.
* Standlone application / deployment in an app container.

Non – Functional Requirement :

Security

* This application have a publicly accessible privacy policy that comprehensively discloses the access,collection,useand sharing of personal and sensitive user data.
* This application create a specifically for the covid-19 response may not access personal and sensitive data that is not required to directly support the public health emergency , and may only use the data collected to support covid-19 releated efforts.
* You must disclose the use of covid -19 relateddata in user -facing privacy disclosures (e.g.,privacy policy and/or in application disclousers) for application
* This application handle all personal or sensitive user data securely , including transmitting it using modern cryptography (for example , over HTTPS).

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 organization.

Logging :

* The system supports logging (app/web/DB).

Cloud :

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

Browser Compatible :

* All latest browsers.

Technology stack :

* HTML & CSS.
* Java Script.

Key points to remember :

1. The ID and attributes 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, coding conventions, method definitions and return types adhere strictly to the endpoints given below.

Application assumption :

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.

Validation :

* Basic mobile validation should be performed.
* Aadhar password validation should be performed.

Project tasks :

* Complete the “COVID-19 contact tracing and status” section in the App content page.
* Submit proof of eligibility via the Advance Notice form.
* Privacy requirements.
* App visibility and user awareness.

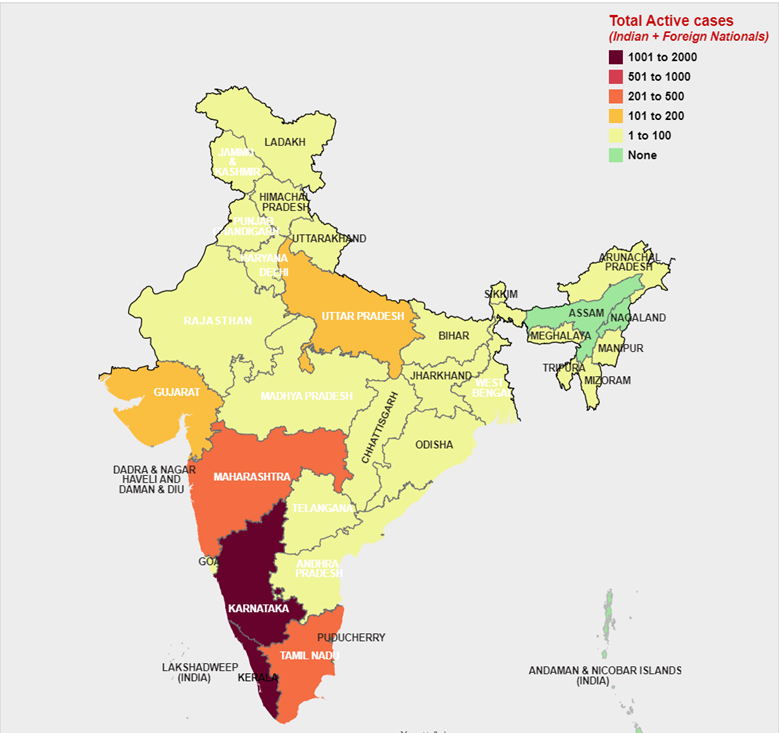
o For apps that collect information in the foreground or use foreground service.

o For apps that collect information when running as a background service.

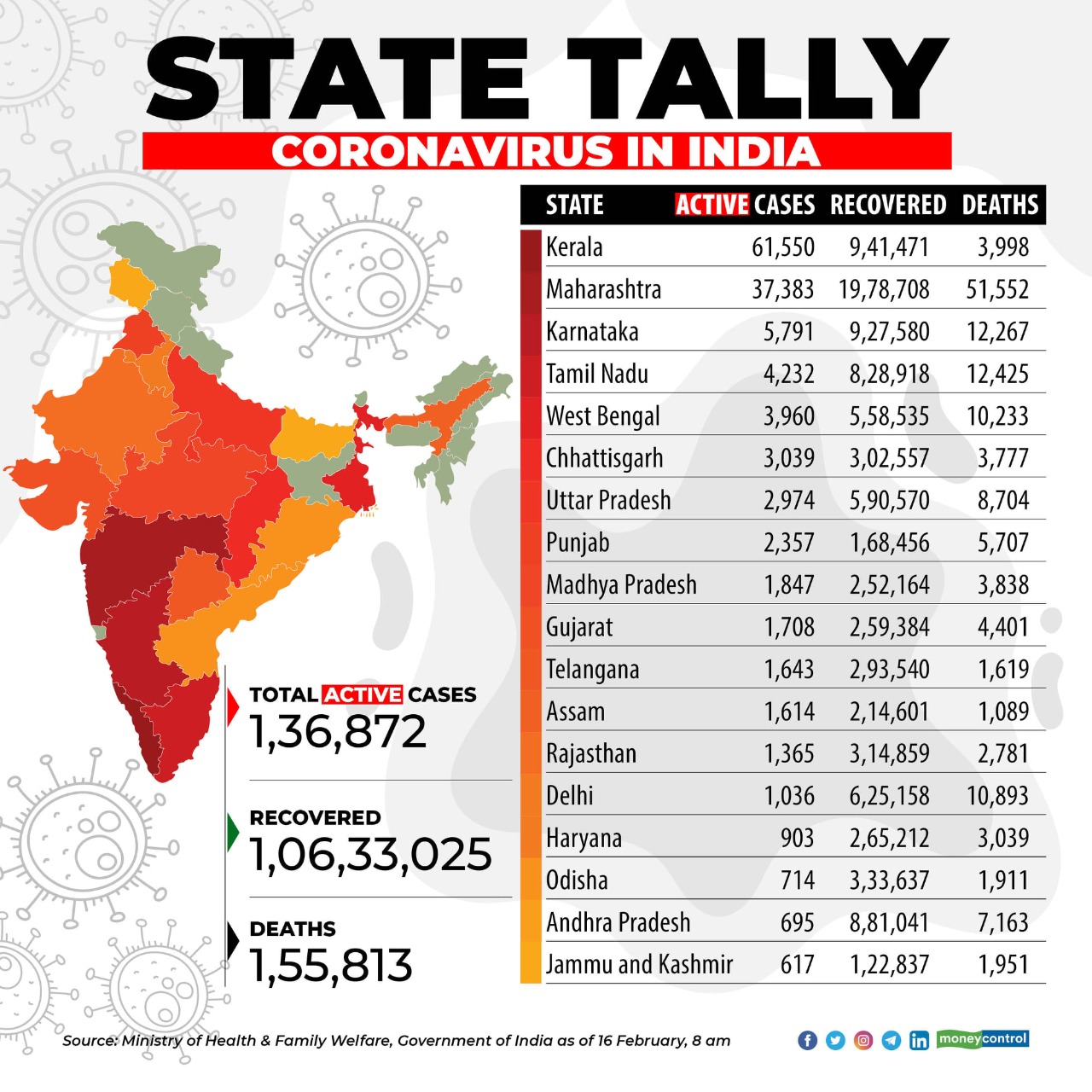
* API requirements.
* Editorial and quality requirements.
* App review and visibility.

User :

* Covid-19 heat map
* State wise covid – 19 cases in india.



State wise covid-19 cases



Covid – 19 cases distric wise

Frontend :

Costomer :

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

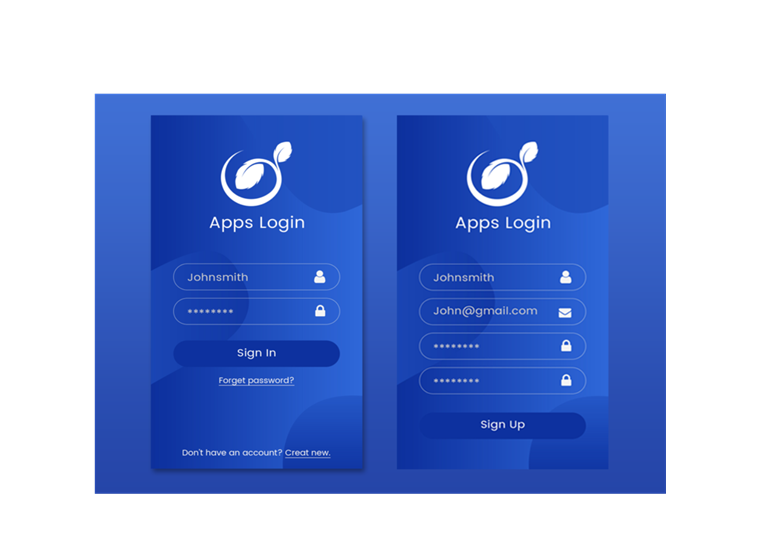
existing customer can log in using the registered email id and password.

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 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. userNavbar

2. HomeButton

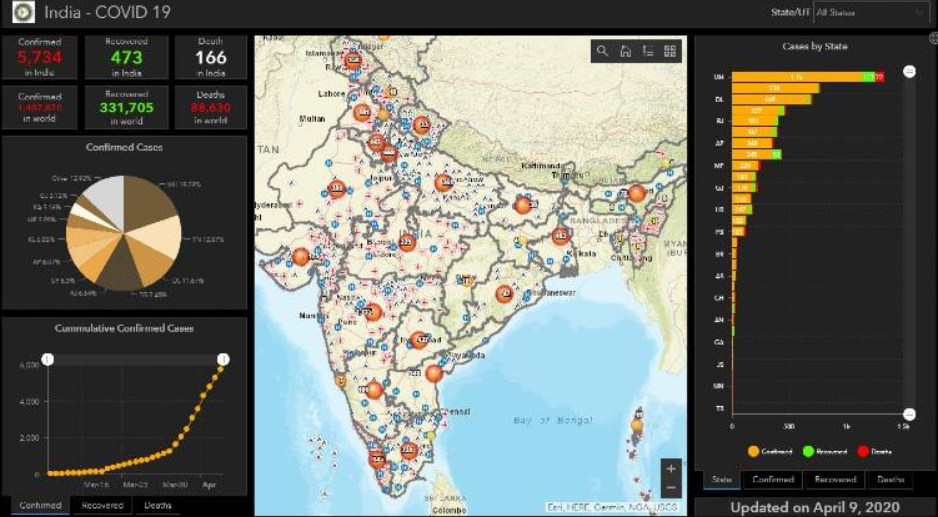
3. Personnel data

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

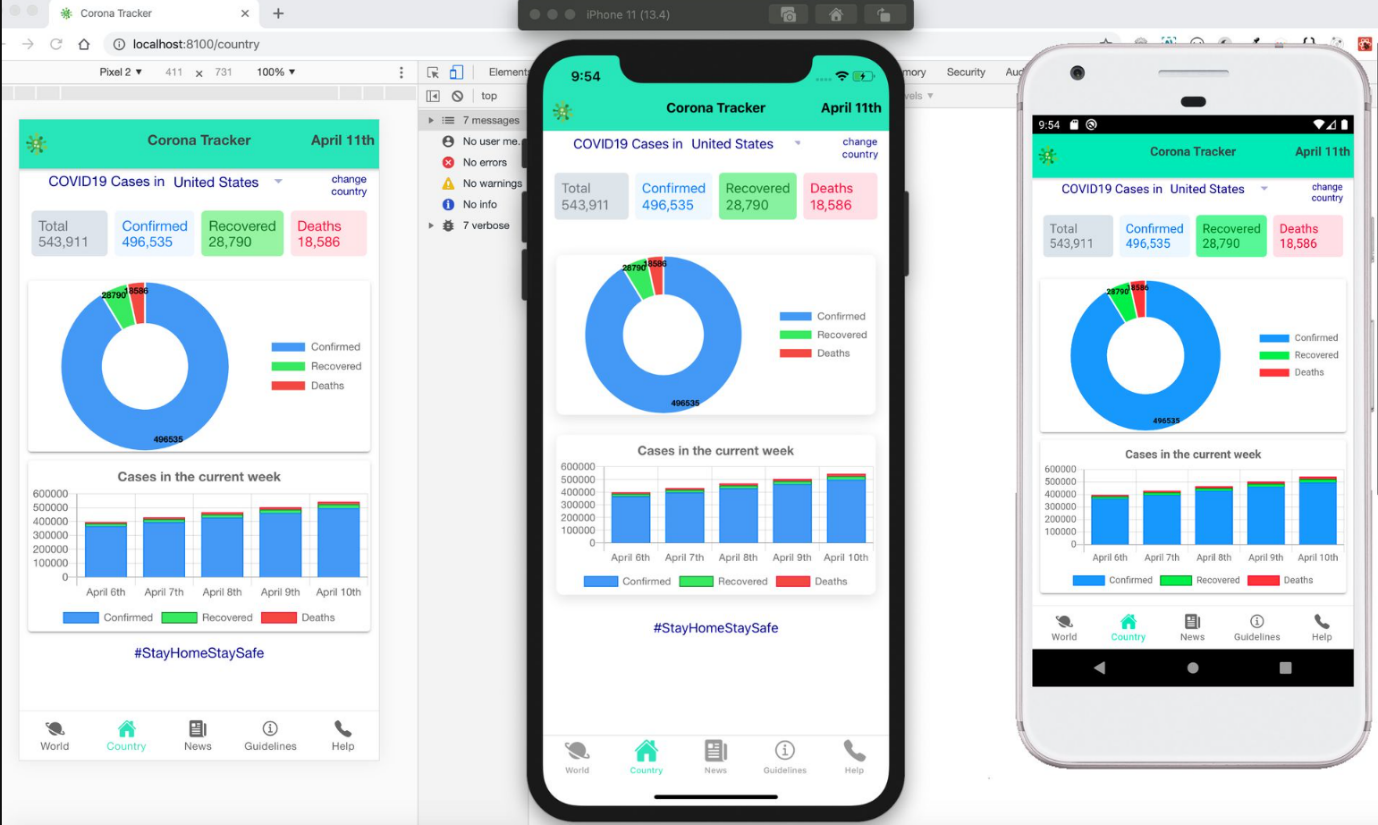
5.affected

6.Recovered

7.adddataButton

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

c.Screenshot:



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 data base.