**R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution)**

RSM NAGAR, PUDUVOYAL-601206

**COVID 19 TRACKER APPLICATION**

**Objective:**

Covid 19 Tracker is an application to track number of people infected with Covid.

**Users of the System:**

1.   Admin

2.   Public

**Functional Requirements:**

* Build an application that users can access and track the number of people infected by Covid online.
* The application should have signup, login, profile, updated status as graph.
* This application should have a provision to maintain a database for Confirmed cases, Active cases and Recovered cases.
* Also, an integrated platform required for admin and patients.
* Administration module to include options for adding / modifying / removing the existing information about affected cases and patient management.
* **Every day the details must be updated.**

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

* Filters to show which district is most affected on that particular day, Districts in which the number of people getting affected reduces / increases.
* SMS to be sent for intimating the neighbours of affected persons.
* Multi-factor authentication for the sign-in process
* Pictorial representation of the number of people affected (both old and new case) and cured.

**Output/Post Condition:**

* Records in database to be represented as a graphical image
* Webpage deployed in a server

**Non-Functional Requirements:**

**Security**

* Webpage –User Name/Password-Based Credentials
* Sensitive data has to be categorized and stored in a secure manner
* Secure connection for transmission of any data

**Performance**

Peak Load Performance (during Winter season, Festival days, National holidays etc)

* Accessibility -< 3 Sec
* Admin application < 2 Sec

Non-Peak Load Performance

* Accessibility < 2 Sec
* Admin Application < 2 Sec

**Availability**

       99.99 % Availability

**Standard Features**

* Scalability
* Maintainability
* Usability
* Availability
* Reliability
* Security

**Logging & Auditing**

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

**Monitoring**

* Should be able to monitor the statistical representations.

**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

**Technology Stack**

* **Front End** HTML/CSS/JAVASCRIPT
* **Database** MySQL

**Key points to remember:**

1.   The patient 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 framework (Folder structure), coding conventions, method definitions and return types.

4. Adhere strictly to the endpoints given below.

**Application assumptions:**

1.   The login page should be the first page rendered when the application loads.

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 as the username and password to navigate to the admin dashboard.

**Validations:**

1.   Basic email validation should be performed.

2.   Basic mobile validation should be performed.

3. Password validations should be performed

**Project Tasks:**

**API Endpoints:**

**USER**

**Action URL Method Response**

1. Register / Login
2. Fetch State -> District -> Place
3. Get details about the tested cases, positive cases, death rate, number of hospitalizations and number recovered
4. View the statistical data
5. Fetch patient health status and provide remedial measures
6. Logout

**ADMIN**

**Action URL Method Response**

1. Get All details about the tested cases, positive cases, death rate, number of hospitalizations and number recovered
2. Add new tested cases, positive cases, death rate, number of hospitalizations and number recovered
3. Display the statistical information

**Frontend:**

**Customer:**

1. **Authentication**: Design an authentication component where the customer can authenticate login and signup credentials

2.   **Signup**: Design a signup page component where the new customer has options to sign up by providing their basic details.

Successful completion of signup process provides authentication to the user to log into the webpage.

Basic details:

* Email
* Username
* Mobile number
* Password
* Confirm password
* Submit Button
* Signin Link
* Signup Box

3.   **Login**: Design a login page component where the authenticated customer can log in using the registered email id and password.

* Email
* Password
* Submit Button
* Signup Link
* Login Box

4.   **Dashboard / Home**: Design a home page component that has the search bar for selecting the district or state, a component to display the date and time of updation and four components to display dynamic statistical graph.

* search bar
* Date and time
* Statistical graph – Positive case, Hospitalized, Recovered and Death rate
* A table to display the number of tested cases, positive cases, death rate, number of hospitalizations, number recovered and number of people vaccinated
* Logout Box

**Admin:**

**Admin Dashboard:** Design a dashboard page where the database has been frequently updated with the number of tested cases, positive cases, death rate, number of hospitalizations and number recovered on the admin side.

**ER Diagram**

