

PROFESSIONAL TRAINING REPORT
At
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By
SHARIN SATHIQ
REG. NO. 39120108



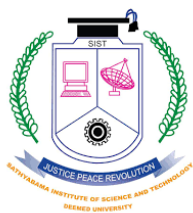
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SHARIN SATHIQ** (Reg. No: 39120108) who carried out the project entitled “**covid19Tracker using Angular**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr. G.MADHIVANAN

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SHARIN SATHIQ** hereby declare that the project report entitled **covid19Tracker using Angular** done by me under the guidance of **Dr.G.MADHIVANAN** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

**PLACE:
CANDIDATE**

SIGNATURE OF THE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G.MADHIVANAN** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

COVID 19 TRACKING application is to track a covid cases in Tamilnadu district. This system is a. This system provides a solution JavaScript Frameworks like Angular, Nodejs,express and database like MONGODB.

Angular is a platform and framework for building single page application using html and typescript

It implement core and optional functionality as a set of typescript libraries that we import into our application the architecture the architecture of an angular application relies on certain fundamental concept .The basic building blocks are the Ng modules in this project many component are created and routes are been set up between each component for the dynamic motion of one page to another page .In this covid tracker app the user can view the details of the affected patient by covid19 the admin can login into the admin can login into the admin panel and add,edit and detail and can logout.Another feature of the app is admin user can search the patient detail

For the backend the app is connected to the mongodb so that information of the patient will be stored and display

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	LIST OF ABBREVIATIONS	iv
1.	INTRODUCTION	1
	WHAT IS ANGULAR JS?	1
	ROLE OF HTML, CSS & JS IN ANGULAR	2
	REQUIREMENTS FOR ANGULAR PROJECT	3
2.	COVID TRACKER APP	4
	OVERVIEW	4
	FUNCTIONAL REQUIREMENTS	4
	COMPONENTS	5
	OVERVIEW OF COMPONENTS	5
	BACKEND	16
3.	SUMMARY AND CONCLUSIONS	17
	SUMMARY	17
	CONCLUSIONS	17
	REFERENCES	18
	APPENDIX	18
	A. SOURCE CODE	18
	B. SCREENSHOTS	42

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	Login Page	6
1.2	Register Page	7
1.3	user patient detail	8
1.4	login Admin page	9
1.5	Terms and Conditions Page	10
1.6	User Management Page	11
1.6.1	Add patient detail	12
1.6.2	Edit patient detail	13
1.6.4	Delete patient detail	14
1.7	Logout Page	15
1.8	Mongo db connection	16

LIST OF TABLES

TABLE No.	TITLE	PAGE No.
2.1	Login Fields	6
2.2	Register/Sign-up Fields	7
2.3	user patient detail	8
2.4	login Admin page	9
2.5	Add patient detail	11
2.6	Edit patient detail	12

LIST OF ABBREVIATIONS

Mongodb	-	Mongo database
App	-	Application
Chars	-	Characters
No's	-	Numbers
Acc	-	According
JS	-	Java Script
Const	-	constant

INTRODUCTION

What is Angular Js?

The AngularJS framework works by first reading the Hypertext Mark-up Language (HTML) page, which has additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit Document Object Model (DOM) manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

- to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- to decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.
- to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

AngularJS implements the MVC pattern to separate presentation, data, and logic components. Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced. AngularJS directives allow the developer to specify custom and reusable HTML-like elements and attributes that define data bindings and the behaviour of presentation components. Some of the most commonly used directives are:

Since **ng-*** attributes are not valid in HTML specifications, **data-ng-*** can also be used as a prefix. For example, both **ng-app** and **data-ng-app** are valid in AngularJS.

ng-app

Declares the root element of an AngularJS application, under which directives can be used to declare bindings and define behavior.

ng-aria

A module for accessibility support of common ARIA attributes.

ng-bind

Sets the text of a DOM element to the value of an expression. For example, `<span ng-bind="name"></span>` displays the value of 'name' inside the span element. Any change to the variable 'name' in the application's scope reflect instantly in the DOM.

ng-class

Conditionally apply a class, depending on the value of a boolean expression.

ng-controller

Specifies a JavaScript controller class that evaluates HTML expressions.

ng-if

Basic if statement directive that instantiates the following element if the conditions are true. When the condition is false, the element is removed from the DOM. When true, a clone of the compiled element is re-inserted.

ng-init

Called once when the element is initialized.

ng-model

Similar to **ng-bind**, but establishes a two-way data binding between the view and the scope.

Role of HTML, CSS & JS in Angular:

- A webpage can contain headings, paragraphs, images, videos, and many other types of data. Front end developers use the HTML element to specify what kind of information each item on a webpage contains — for instance, the “p” HTML element indicates a paragraph. Developers also write HTML language code to specify how different items relate to one another in the page’s overall structure or document structure.
- CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. ... CSS helps us to control the text color, font style, the spacing between paragraphs, sizing of columns, layout designs, and many more. CSS provides these capabilities through a technical model based on a hierarchical scope of effect, the separation of style from content, and a well-defined set of published standards.
- JavaScript is the most popular and most widely developed programming language ever existed. JavaScript is a must-have for Web developers, Front-end/ Back-end developers, Mobile / Desktop developers, Graphics / Game developers and even for Full Stack developers as well.
- Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.
- AngularJS provides some amazing capabilities out of the box, and can convert your non-trivial jQuery based apps into something more powerful and yet more simple to maintain. Some frameworks simply bundle together the existing tools which make app development very difficult. However, Angular was carefully engineered to ensure that every tool works optimally and delivers great results.
- Developers find Angular very effective in terms of creating dynamic, single page application, and supporting MVC(Model-View-Controller) programming structure. At a present span of time, in the competitive business environment, Angular has emerged as the superhero due to its intuitiveness and rich feature

Requirements of Angular Project:

- Install Angular CLI
- Create a workspace and initial application
- Run the application

Modules, components and services are classes that use *decorators*. These decorators mark their type and provide metadata that tells Angular how to use them.

- The metadata for a component class associates it with a *template* that defines a view. A template combines ordinary HTML with Angular *directives* and *binding mark-up* that allow Angular to modify the HTML before rendering it for display.
- The metadata for a service class provides the information Angular needs to make it available to components through *dependency injection (DI)*.

Angular *NgModules* differ from and complement JavaScript (ES2015) modules. An *NgModule* declares a compilation context for a set of components that is dedicated to an application domain, a workflow, or a closely related set of capabilities. An *NgModule* can associate its components with related code. Every Angular application has at least one component, the *root component* that connects a component hierarchy with the page document object model (DOM). Each component defines a class that contains application data and logic, and is associated with an HTML *template* that defines a view to be displayed in a target environment.

The `@Component()` decorator identifies the class immediately below it as a component, and provides the template and related component-specific metadata.

A template combines HTML with Angular mark-up that can modify HTML elements before they are displayed. Template *directives* provide program logic, and *binding mark-up* connects your application data and the DOM.

COVID TRACKING APPLICATION

Overview:

This project aims to maintain the information about the patients those who are affected by covid 19. The admin will manage details of the patients. Admin will be able to login themselves and add data of the of the patients and for the admin purpose, user management portal will be also available.

Only admin will be able to update patients information. There are many components like login, patient details etc. Once user register themselves, they will be directed to various fields . This app is done with HTML, CSS, JS and the main part is the angular js.

The Covid tracker App designed by Team Amethyst will be very easy and a delicate for the user to view the details of affected patients. Now a days, internet is the most widely used network in modern world , For their easy purpose, Team Amethyst made this covid tracker App, so that users can acknowledge more information about the pandemic around them.

Functional Requirements:

To build an angular application, the main thing needed is the components where are those application routing process occurs. The components used in this application are listed below:

COMPONENTS:

- ❖ Login
- ❖ Register
- ❖ Patients details
- ❖ Admin login
- ❖ User Management Actions (for admin purpose)
 - Add
 - Edit
 - Delete
- ❖ Logout

OVERVIEW OF COMPONENTS:

- Login portal:

The login portal is used for the users to login into the app. The admin also can login to do the user management system.

FIELDS	VALIDATIONS
useremail	Required
Password	Required

Table 2.1: Login Fields

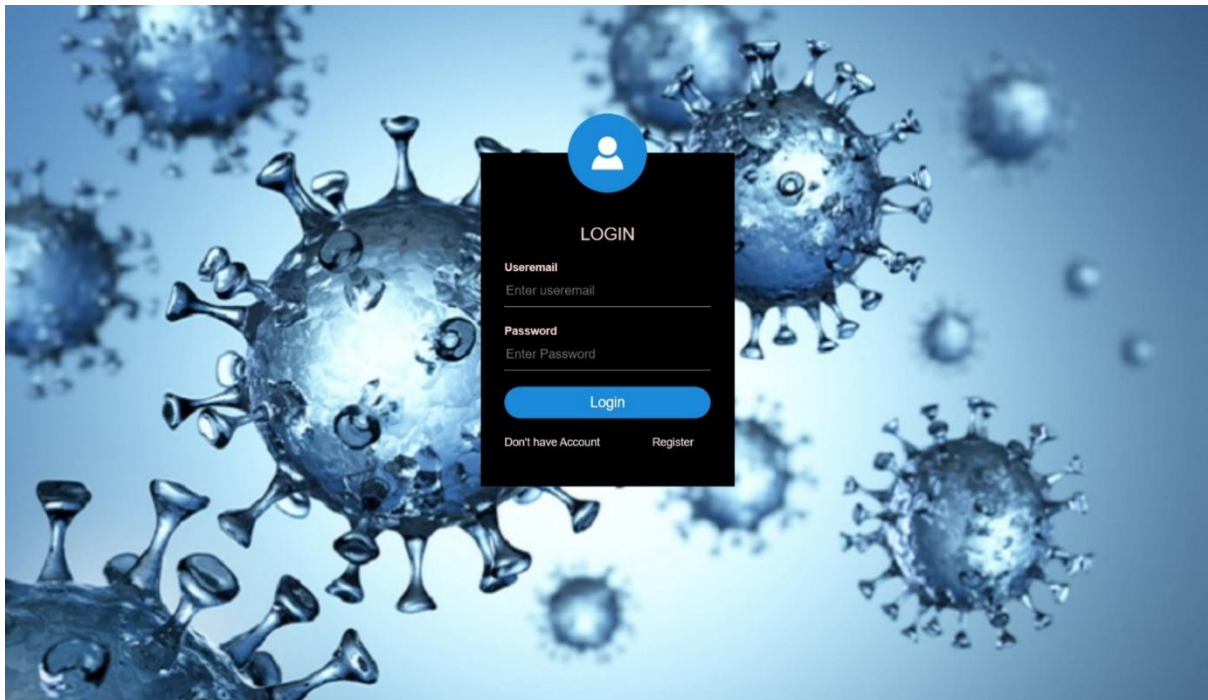


Fig 1.1: Login Page

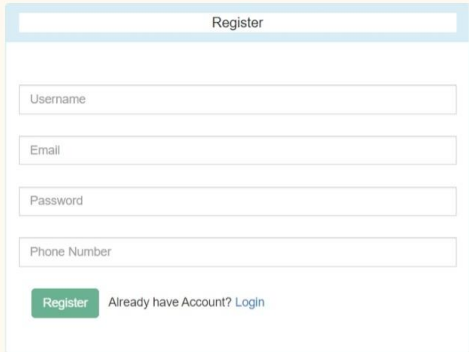
- Register:

User will be able to register themselves by clicking register link from login

screen. On submission, data gets inserted to the database via mongodb.

FIELDS	VALIDATIONS
Username	Required
E-mail	Required, E-mail validation
Password	
Phone no	Number validation,

Table 2.2: Register



The screenshot displays a 'Register' form on a light orange background. The form is a white box with a light blue header containing the title 'Register'. It features four input fields: 'Username', 'Email', 'Password', and 'Phone Number'. Below the fields, there is a green 'Register' button and a link that says 'Already have Account? Login'.

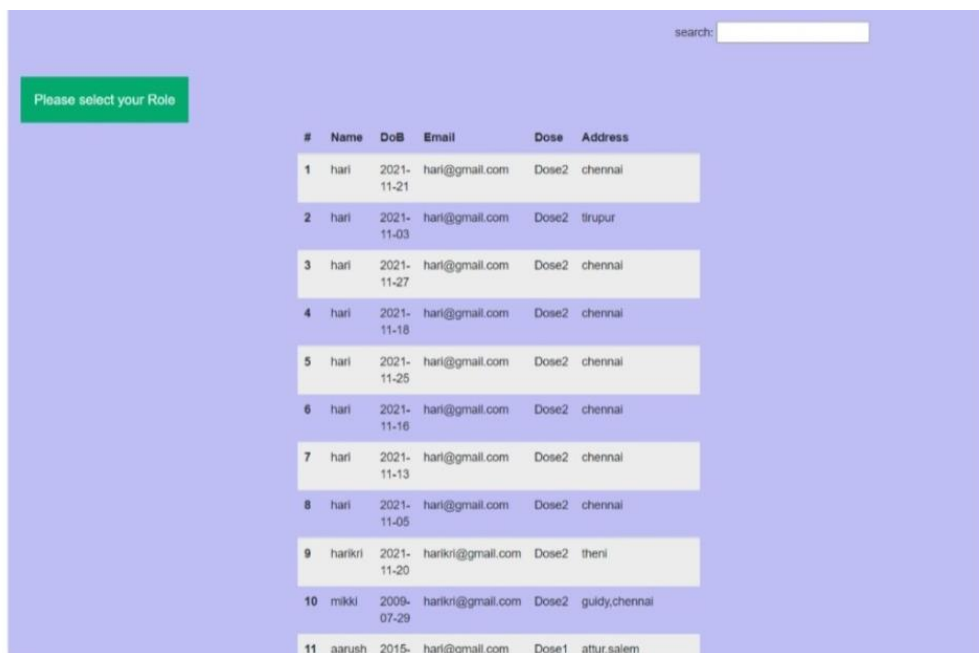
Fig 1.2: Register Page

- Patient details:

In patients detail portal, the users can view the affected patients details

FIELDS	VALIDATIONS
Name	Required
DOB	Required
Email	Required
Dose	Required
Address	Required

Table 2.3: Patients details

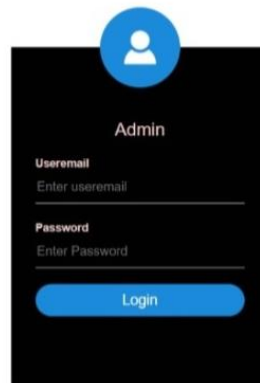


#	Name	DoB	Email	Dose	Address
1	hari	2021-11-21	hari@gmail.com	Dose2	chennai
2	hari	2021-11-03	hari@gmail.com	Dose2	tirupur
3	hari	2021-11-27	hari@gmail.com	Dose2	chennai
4	hari	2021-11-18	hari@gmail.com	Dose2	chennai
5	hari	2021-11-25	hari@gmail.com	Dose2	chennai
6	hari	2021-11-16	hari@gmail.com	Dose2	chennai
7	hari	2021-11-13	hari@gmail.com	Dose2	chennai
8	hari	2021-11-05	hari@gmail.com	Dose2	chennai
9	harikri	2021-11-20	harikri@gmail.com	Dose2	theni
10	mskki	2009-07-29	harikri@gmail.com	Dose2	gudiy, chennai
11	aarush	2015-	hari@gmail.com	Dose1	attur, saiem

Fig 1.3: Patients details

- Admin portal :

Allows the access to admin panel .

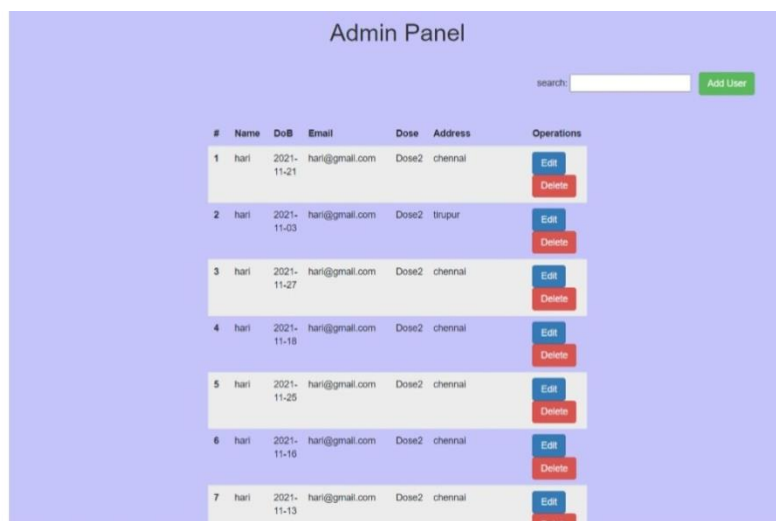


A dark-themed login form for the Admin portal. At the top is a blue circular icon with a white person silhouette. Below it, the word "Admin" is centered. The form contains two input fields: "Useremail" with the placeholder "Enter useremail" and "Password" with the placeholder "Enter Password". A blue "Login" button is at the bottom.

Fig 1.4: admin panel

- User Management Portal:

This portal is fully for the purpose of admin in-order to add , edit , delete the details from the management portal.



A screenshot of the "Admin Panel" for user management. It features a search bar and an "Add User" button at the top right. Below is a table with 7 rows of user data. Each row has columns for ID, Name, DoB, Email, Dose, Address, and Operations (Edit/Delete buttons).

#	Name	DoB	Email	Dose	Address	Operations
1	hari	2021-11-21	hari@gmail.com	Dose2	chennai	Edit Delete
2	hari	2021-11-03	hari@gmail.com	Dose2	tripur	Edit Delete
3	hari	2021-11-27	hari@gmail.com	Dose2	chennai	Edit Delete
4	hari	2021-11-18	hari@gmail.com	Dose2	chennai	Edit Delete
5	hari	2021-11-25	hari@gmail.com	Dose2	chennai	Edit Delete
6	hari	2021-11-16	hari@gmail.com	Dose2	chennai	Edit Delete
7	hari	2021-11-13	hari@gmail.com	Dose2	chennai	Edit Delete

Fig 1.6: User Management Page

- There are certain actions like add , edit and delete info from database.

- Add patients Portal:

Add patients component mainly involves the admin action of adding player apart from user registration.

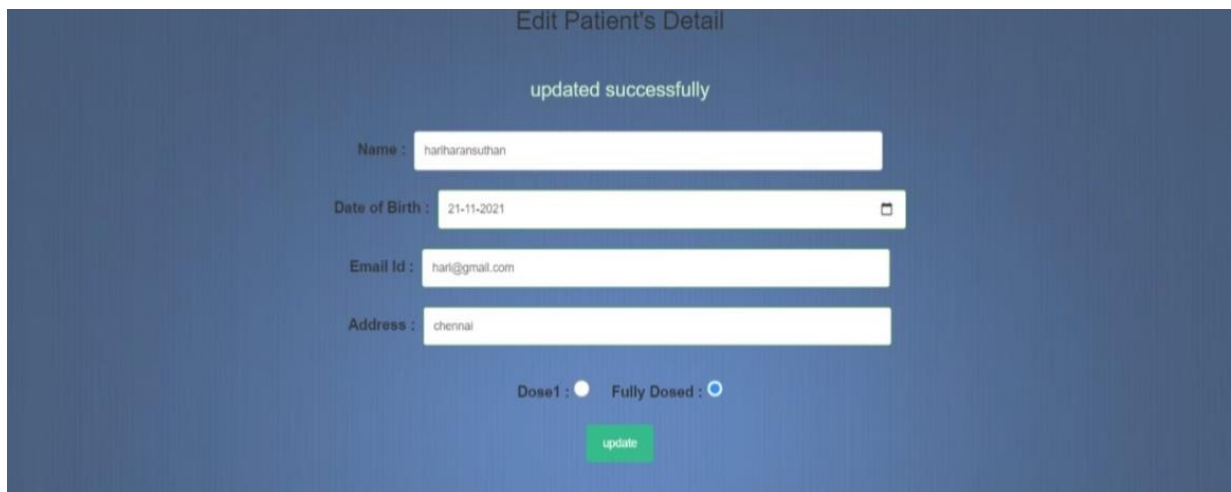
FIELDS	VALIDATIONS
Username	Required, Minlength = 6
E-mail	Required, E-mail validation
DOB	Required
Address	Required

Table 2.5: Add Patients Fields

Fig 1.6.1: Add Patients Page

- **Edit patients Portal:**

The user management portal contains a button named edit in side of every registered patient name. On clicking that button the admin gets directed to the edit patients portal where all the data of patients' already entered will be available. Thereby admin can update the patients' information and the database also gets updated.



The screenshot displays the 'Edit Patient's Detail' interface. At the top, it says 'updated successfully'. Below this, there are four input fields: 'Name' with the value 'hariharansuthan', 'Date of Birth' with '21-11-2021' and a calendar icon, 'Email Id' with 'hari@gmail.com', and 'Address' with 'chennai'. At the bottom, there are two radio buttons: 'Dose1' (unselected) and 'Fully Dosed' (selected). A green 'update' button is positioned below the radio buttons.

Fig 1.6.2: Edit Patients portal

- **Delete User Option:**

Same like edit option, there exists a delete button in side of every patients data. On clicking that delete button, a dialogue box appears asking that sure deletion should be done or not. As per the admin wish, the deletion operation will be done.

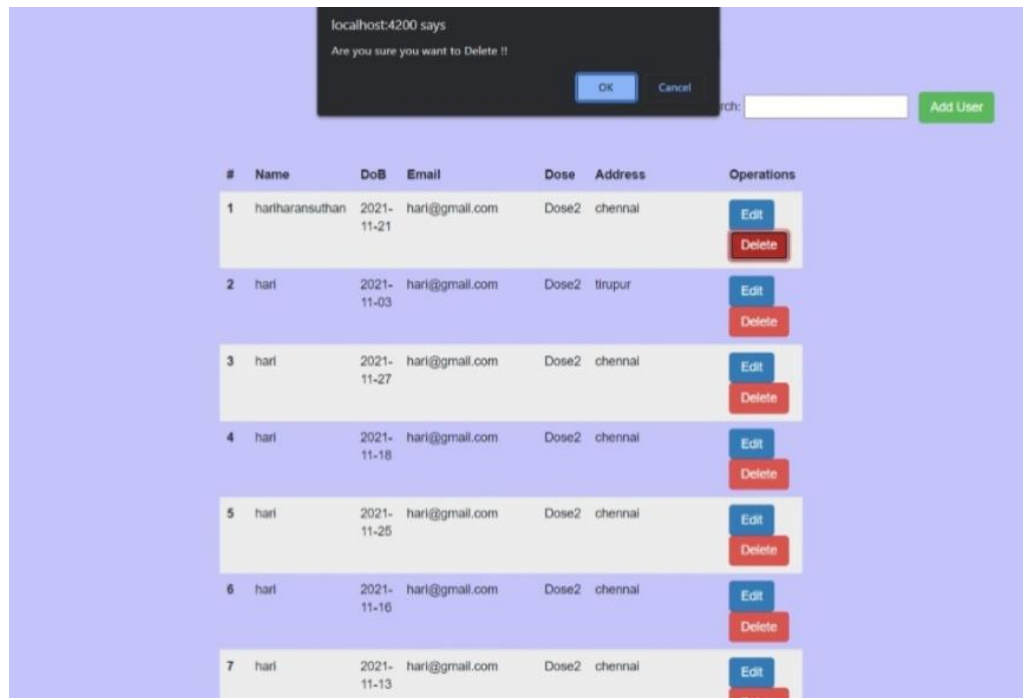


Fig 1.6.4: Delete Player Page

- Logout:

After updating everything ,the info will be updaed in user portal .once the updation is completed the admin can logout from the admin portal. this is the final stage of our application.

- **BACKEND:**

The backend duties involve the mongodb connection and a service. For the backend actions, a java script document is done in backend and thereby the mongodb cluster is connected to the application.

For this to be carried out, first of all a mongodb cluster should be created and a database should be made and thereby it should be connected with backend through the mongodb database connection link.

A service named user service is imported to carry out all the backend function and call back functions made to get the responses of the server.

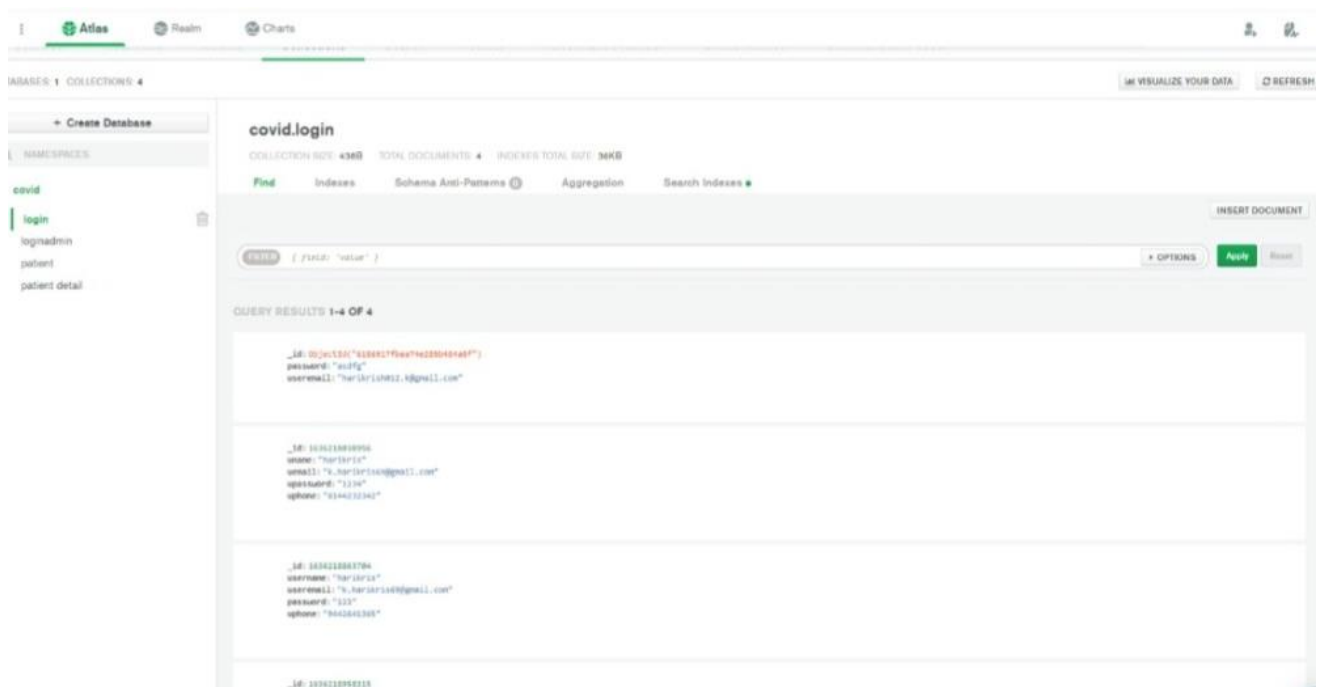


Fig 1.8: Mongodb Connection

SUMMARY AND CONCLUSIONS

- SUMMARY:

This COVID19 TRACKING app helps many people those who're more HELP in this pandemic situation. Now a days in the covid situation this app will us to know how many people are suffering and what is the news about that and getting help from doctor in the related activity is important. Therefore I created "COVID 19 TRACKING APP" in which the PATIENT will be done is fast manner and in an This app is a single page web application build with Angular JS. More technical and database based activities are carried out to build this app. The code is done from the scratch and the layout is made. The data is passed through database and collected in mongodb. This will be more helpful in tracking !!!

- CONCLUSION:

With the steps shown in the above section, we can make lot of single page applications that helps in the modern technology. Angular js makes the building of angular application easier as compare to many application languages. Working with angular gives more knowledge in importing modules and dealing with routes. It also involves processes in backend and the connection between the application and mongodb database also exists.

Angular is now an emerging technology which helps the full stack developers in developing their own application.

BACKEND:

```
const exp =require("express");
const cors=require("cors");
const bodyparser = require("body-parser");
const mongodb =require("mongodb").MongoClient;
const { decodedTextSpanIntersects } = require("typescript");
const app=exp();
```

```

app.use(cors());
app.use(bodyParser.json());

var db;
mongodb.connect("mongodb+srv://hari:8144659365@cluster0.azr0g.mongodb.net/covid?retryWrites=true&w=majority",(error,result)=>{
if(error){
    console.log("DB not connected!!!");
}
else{
    db=result.db("covid")
    console.log("db created!!");
}
});
app.use((req,res,next)=>{

console.log("MID1")

    next();
});

// app.use((req,res,next)=>{

//     console.log("MId2")

//     next();
// });

//     function veri(req,res,next)
//     {
//         console.log("user verified");
//         next();
//     }
// app.get("/",(req,res)=>{

// console.log("home");

// res.send("<h1>Express</h1>");

// });

// app.get("/contact",veri,(req,res)=>{
//     console.log("contact");
//     res.json("welcome to com=ntact");
// });

```

```

app.post("/contact", (req, res) => {

    req.body._id = new Date().getTime();
    console.log(req.body);

    db.collection("patient").insertOne(req.body, (error, data) => {
        if (error) {
            res.status(403).json("ERROR!!!");
        } else {
            res.json("User Resgitered successfully");
        }
    });
});

app.post("/patient", (req, res) => {

    req.body._id = new Date().getTime();
    console.log(req.body);

    db.collection("patient detail").insertOne(req.body, (error, data) => {
        if (error) {
            res.status(403).json("ERROR!!!");
        } else {
            res.json("User Resgitered successfully");
        }
    });
});

app.post("/login", (req, res) => {

    console.log(req.body);

    db.collection("login").find(req.body, { projection: { _id: 1, useremail: 1 } }).toArray((error, data) => {

        if (error) {
            {
                res.status(403).json("Error in Finding the login doc");
            }
        } else {
            {
                res.json(data);
            }
        }
    });
});
});

```

```

app.post("/register", (req, res) => {

    req.body._id = new Date().getTime();
    console.log(req.body);

    db.collection("login").insertOne(req.body, (error, data) => {
        if (error) {
            res.status(403).json("ERROR!!!");
        } else {
            res.json("User Resgitered successfully");
        }
    });
});

app.get("/adduser", (req, res) => {

    db.collection("patient detail").find().toArray((error, data) => {

        if (error)
        {
            res.status(403).json("Error in Finding the login doc");
        }
        else {
            res.json(data);
        }
    });
});

app.get("/unamecheck/:Name", (req, res) => {

    console.log(req.params.Name);
    db.collection("patient
detail").find({Name: req.params.Name}, {projection: {_id: 1}}).toArray((error, data)
=> {

        if (error)
        {
            res.status(403).json("Error in Finding the Patient Name
Availability");
        }
        else {
            res.json(data);
        }
    });
});

```

```

});

app.get("/getuser/:userid", (req, res) => {
  console.log(req.params);
  db.collection("patient
detail").find({_id: Number(req.params.userid)}).toArray((error, data) => {

    if (error)
    {
      res.status(403).json("Error in Finding the Patient Name
Availability");
    }
    else {
      res.json(data);
    }
  });
});

app.put("/updateuser", (req, res) => {
  console.log(req.body);

  var con = {_id: req.body._id};
  var
newvalue = { $set: { Name: req.body.Name, email: req.body.email, date: req.body.date, Dose: req.body.Dose, Address: req.body.Address } };
  // res.json("successfully updated");
  db.collection("patient detail").update(con, newvalue, (error, data) => {
    if (error) {
      res.status(403).json("Error in Finding the Patient Name
Availability");
    } else {
      res.json("updated successfully");
    }
  })
});

app.delete("/deleteuser/:userid", (req, res) => {

  console.log(req.params);
  db.collection("patient
detail").deleteOne({_id: Number(req.params.userid)}, (error, data) => {
    res.json("patient detail deleted successfully!!");
  });
});

app.get("/search/:searchtxt?", (req, res) => {
  console.log(req.params);

```

```

    if(req.params.searchtxt != undefined){

        var search=new RegExp(req.params.searchtxt,'i');
        var scond={Name:search};
        }
        else{
            var scond = null;
        }
        db.collection("patient detail").find(scond).toArray((error,data)=>{
            res.json(data);
        });
    });

app.get("/adduseradmin",(req,res)=>{

    db.collection("patient detail").find().toArray((error, data)=>{

        if(error)
        {
            res.status(403).json("Error in Finding the login doc");
        }
        else {
            res.json(data);
        }
    });

});

app.get("/searc/:searchtx?",(req,res)=>{
    console.log(req.params);
    if(req.params.searchtx != undefined){

        var searc=new RegExp(req.params.searchtx,'i');
        var scond={Name:searc};
        }
        else{
            var scond = null;
        }
        db.collection("patient detail").find(scond).toArray((error,data)=>{
            res.json(data);
        });
    });

});

app.post("/loginadmin", (req, res)=>{

    console.log(req.body);

```

```

    db.collection("loginadmin").find(req.body,{projection:{_id:1,useremail:1}}
).toArray((error, data)=>{

    if(error)
    {
        res.status(403).json("Error in Finding the login doc");
    }
    else {
        res.json(data);
    }
});

});

module.exports=app;

```

Appuser.component

```

.dropbtn {
    background-color: #04AA6D;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
}

.dropdown {
    position: relative;
    display: inline-block;
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}

.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
}

```

```

    display: block;
}

.dropdown-content a:hover {background-color: #ddd;}

.dropdown:hover .dropdown-content {display: block;}

.dropdown:hover .dropbtn {background-color: #3e8e41;}

```

```

<div class="container" style="background-color:rgb(196, 196, 250)">
  <h1 style="text-align: center;">Admin Panel</h1>
  <div class="row" style="padding-top: 30px;">
    <div class="col-sm-4"></div>

    <div class="col-sm-4"></div>

    <div class="col-sm-4">
      search: <input type="text" name="search"
#search (keyup)="dosearch(search.value)"> &nbsp;
      <a routerLink="/patient's details"> <button class="btn btn-
success">Add User</button></a>
    </div>
  </div>
  <br><br>

  <div class="row">
<div class="col"></div>

<div class="col">
  {{msg}}
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Name</th>
        <th scope="col">DoB</th>
        <th scope="col">Email</th>
        <th scope="col">Dose</th>
        <th scope="col">Address</th>
        <th scope="col">Operations</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let user of users; index as i;">
        <th scope="row">{{i+1}}</th>
        <td>{{user.Name}}</td>
        <td>{{user.date}}</td>

```

```

        <td>{{user.email}}</td>
        <td>{{user.Dose}}</td>
        <td>{{user.Address}}</td>
        <td>
            <a routerLink="/editpatient/{{user._id}}"> <button class="btn
btn-primary">Edit</button></a> &nbsp;
            <button class="btn btn-
danger"(click)="deleteuser(user._id)">Delete</button>
        </td>
    </tr>
    <p *ngIf="users.length==0" style="text-align: center;"> No Records
Found :(</p>
    </tbody>
</table>
</div>

<div class="col"></div>
</div>
</div>

import { Component, OnInit } from '@angular/core';

import { ContactService } from '../contact.service';

@Component({
  selector: 'app-adduser',
  templateUrl: './adduser.component.html',
  styleUrls: ['./adduser.component.css']
})
export class AdduserComponent implements OnInit {
  users:any[]=[];
  msg:string;
  constructor(public user:ContactService) { }

  ngOnInit(): void {

    this.user.getpatient().subscribe((data:any[])=>{
      console.log(data);
      this.users=data;
    },(error:any)=>{
      console.log(error);
      this.msg="Something went wrong";
    })
  }
  deleteuser(userid:number){
    if(confirm("Are you sure you want to Delete !! ")){

```

```

    this.user.deleteuserdata(userid).subscribe((data:string)=>{
      this.msg=data;
      var index=this.users.findIndex((obj)=>{
        return obj._id == userid;
      });
      this.users.splice(index,1);
    },(error:any)=>{
      console.log(error);
      this.msg="Something went wrong!";
    });
  }
}

dosearch(search:string){
  this.user.searchuser(search).subscribe((data:any[]=>{
    console.log(data);
    this.users=data;
  },(error:any)=>{
    console.log(error);
  }));
}
}
}

```

AdduserAdmin

```

.dropbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
}

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
}

```

```

padding: 12px 16px;
text-decoration: none;
display: block;
}

.dropdown-content a:hover {background-color: #ddd;}

.dropdown:hover .dropdown-content {display: block;}

.dropdown:hover .dropbtn {background-color: #3e8e41;}
.dropbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
}

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content a:hover {background-color: #ddd;}

.dropdown:hover .dropdown-content {display: block;}

.dropdown:hover .dropbtn {background-color: #3e8e41;}
<div class="container" style="background-color:rgb(190, 190, 245);">
  <div class="row" style="padding-top: 30px;">
    <div class="col-sm-4"></div>

```

```

<div class="col-sm-4"></div>

<div class="col-sm-4">
    search: <input type="text" name="search"
#searc (keyup)="dosearch(searc.value)"> &nbsp;

</div>
</div>
<br><br>
<div class="dropdown">
    <button class="dropbtn">Please select your Role</button>
    <div class="dropdown-content">
        <a routerLink="/loginadmin">Admin</a>
        <a routerLink="/adduseradmin">Login as user</a>

    </div>
</div>

<div class="row">
<div class="col"></div>

<div class="col">
    {{msg}}
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Name</th>
                <th scope="col">DoB</th>
                <th scope="col">Email</th>
                <th scope="col">Dose</th>
                <th scope="col">Address</th>

            </tr>
        </thead>
        <tbody>
            <tr *ngFor="let user of users; index as i;">
                <th scope="row">{{i+1}}</th>
                <td>{{user.Name}}</td>
                <td>{{user.date}}</td>
                <td>{{user.email}}</td>
                <td>{{user.Dose}}</td>
                <td>{{user.Address}}</td>

            </tr>
            <p *ngIf="users.length==0" style="text-align: center;"> No Records
Found :(</p>
        </tbody>

```

```

        </table>
</div>

<div class="col"></div>
</div>
</div>

import { Component, OnInit } from '@angular/core';
import { ContactService } from '../contact.service';

@Component({
  selector: 'app-adduseradmin',
  templateUrl: './adduseradmin.component.html',
  styleUrls: ['./adduseradmin.component.css']
})
export class AdduseradminComponent implements OnInit {
  users:any[]=[];
  msg:string
  constructor(public user:ContactService) { }

  ngOnInit(): void {
    this.user.getuse().subscribe((data:any[])=>{
      console.log(data);
      this.users=data;
    },(error:any)=>{
      console.log(error);
      this.msg="Something went wrong";
    })
  }

  dosearch(searc:string){
    this.user.searchuse(searc).subscribe((data:any[])=>{
      console.log(data);
      this.users=data;
    },(error:any)=>{
      console.log(error);
    });
  }
}

```

Contact component

```

input[type=text], select, textarea {
  width:45%;

```

```

padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
margin-top: 6px;
margin-bottom: 16px;

}
input[type=email]{
width:45%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
margin-top: 6px;
margin-bottom: 16px;
}
form{
border-radius:5px;
background-color: azure;
padding: 20px;

}
input.ng-invalid{
border-color:red;
padding: 20px;
}
input.ng-untouched{
border-color:green;
}

button[type=submit] {
background-color: #04AA6D;
color: white;
padding: 12px 20px;
border: none;
border-radius: 4px;
cursor: pointer;
}

button[type=submit]:hover {
background-color: #45a049;
}

.containe {
border-radius: 5px;
text-align: center;

```

```

padding: 20px;
}

.footer-dark {
padding: 50px 0;
color: #f0f9ff;
background-color: #282d32
}

.footer-dark h3 {
margin-top: 0;
margin-bottom: 12px;
font-weight: bold;
font-size: 16px
}

.footer-dark ul {
padding: 0;
list-style: none;
line-height: 1.6;
font-size: 14px;
margin-bottom: 0
}

.footer-dark ul a {
color: inherit;
text-decoration: none;
opacity: 0.6
}

.footer-dark ul a:hover {
opacity: 0.8
}

@media (max-width:767px) {
.footer-dark .item:not(.social) {
text-align: center;
padding-bottom: 20px
}
}

.footer-dark .item.text {
margin-bottom: 36px
}

@media (max-width:767px) {
.footer-dark .item.text {
margin-bottom: 0
}
}

```

```

    }
}

.footer-dark .item.text p {
    opacity: 0.6;
    margin-bottom: 0
}

.footer-dark .item.social {
    text-align: center
}

@media (max-width:991px) {
    .footer-dark .item.social {
        text-align: center;
        margin-top: 20px
    }
}

.footer-dark .item.social>a {
    font-size: 20px;
    width: 36px;
    height: 36px;
    line-height: 36px;
    display: inline-block;
    text-align: center;
    border-radius: 50%;
    box-shadow: 0 0 0 1px rgba(255, 255, 255, 0.4);
    margin: 0 8px;
    color: #fff;
    opacity: 0.75
}

.footer-dark .item.social>a:hover {
    opacity: 0.9
}

.footer-dark .copyright {
    text-align: center;
    padding-top: 24px;
    opacity: 0.3;
    font-size: 13px;
    margin-bottom: 0
}

```

```

<div style="background-color: rgb(174, 174, 238);background-size: 100px; ">
<h2 style="text-align:center;color: rgb(107, 20, 32);">Contact Us</h2>

```

```

<div class="containe">
  <h3 style="color: green;">{{msg}}</h3>
  <form #reg="ngForm" (ngSubmit)="Registers(reg)" >
    <label for="fname"> Name : &nbsp;</label>
    <input type="text" id="fname" name="Name" placeholder="Your name.."
pattern="[a-zA-Z]+$" #Name="ngModel" required ngModel><br>
    <p class="alert alert-danger" *ngIf="Name.errors?.required &&
Name.touched">please enter Name</p>
    <label for="lname">Email Id : &nbsp;</label>
    <input type="email" id="lname" name="email" placeholder="Your email.."
required email #email="ngModel" ngModel><br>
    <p class="alert alert-danger" *ngIf="email.errors?.required &&
email.touched">please enter email Id</p>
    <p class="alert alert-danger" *ngIf="email.errors?.email &&
email.touched">please enter valid Email id</p>
    <label for="lname">Address : &nbsp;</label>
    <input type="text" id="lname" name="Address" placeholder="your Address"
pattern="[a-zA-Z]+$" #country="ngModel" required ngModel><br>
    <p class="alert alert-danger" *ngIf="country.errors?.required &&
country.touched">please enter Address</p>
    <label for="subject" style="margin-right:700px">Subject :</label><br>
    <textarea id="subject" name="subject" placeholder="Write something.."
#subject="ngModel" style="height:200px" pattern="[a-zA-Z0-9]+$" required
ngModel></textarea><br>
    <p class="alert alert-danger" *ngIf="subject.errors?.required &&
subject.touched">please enter Subject</p>
    <button type="submit" value="Submit"
[disabled]="reg.invalid">Submit</button>
  </form>
</div>
</div>
<br><br>

<div class="footer-dark">
  <footer>
    <div class="container">
      <div class="row">
        <div class="col-md-3 item">
          <h3>Donation</h3>
          <ul>
            <li><a href="#">plasma</a></li>
            <li><a href="#">blood</a></li>
            <li><a href="#">organ</a></li>
          </ul>
        </div>
        <div class="col-md-3 item">
          <h3>Services</h3>
          <ul>

```

```

        <li><a href="#">Download cerificate </a></li>
        <li><a href="#">verification</a></li>
        <li><a href="#">Hosting</a></li>
    </ul>
</div>
<div class="col-md-3 item">
    <h3>About</h3>
    <ul>
        <li><a href="#">government</a></li>
        <li><a href="#">vaccination</a></li>
        <li><a href="#">Instruction</a></li>
    </ul>
</div>
<div class="col-md-3 item text">
    <h3>cowin.com</h3>
    <p>vaccinate if you are not vaccinated</p>
</div>

    <div class="col item social"><a href="#"><i class="fa fa-
facebook"></i></a><a href="#"><i class="fa fa-twitter"></i></a><a href="#"><i
class="fa fa-youtube"></i></a><a href="#"><i class="fa fa-
instagram"></i></a><a href="#"><i class="fa fa-google"></i></a></div>
    </div>
    <p class="copyright">cowin.com © 2021</p>
</div>
</footer>
</div>

```

```

import { NgForm } from '@angular/forms';
import { Component, OnInit } from '@angular/core';
import { ContactService } from '../contact.service';

@Component({
  selector: 'app-contact',
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.css']
})
export class ContactComponent implements OnInit {

  msg:string;
  constructor(public user:ContactService) { }

  ngOnInit(): void {
  }

  Registers(reg:NgForm){

```

```

console.log(reg.value);

this.user.Registerscon(reg.value).subscribe((data:string)=>{

  console.log(data);

  this.msg = data;

reg.reset();

}, (error:any)=>{

  console.log(error);

  this.msg = "Something Went Wrong";

});

}

}

```

District component

```

button[type=submit] ,a{
  background-color: #04AA6D;
  color: rgb(230, 205, 205);
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button[type=submit]:hover {
  background-color: #45a049;
}

```

```

<button type="submit" value="Go Back"><a routerLink="/">Go Back</a></button>
<div style="background-color: blanchedalmond;">
  <br>
<div class="card" style="width: 24rem;height:20rem;margin-right:
500px;display: table-cell;text-align: center;">
  <div class="card-body">
    <h2 class="card-title">Last updated cases</h2>
    <br>

```

```

        <p class="card-text"></p>
        <div *ngIf="isShowResult">
            <div *ngFor="let user of newdata">
                <h1 style="color: red;">{{user.Lastupdated}}</h1>
                <!-- <h1>{{user.District}}</h1> -->
            </div>
            <br>
        </div>

    </div>
</div>
<div class="card" style="width: 24rem;height:20rem;display: table-cell;text-align:center">
    <div class="card-body">
        <h2 class="card-title">District name</h2>
        <br>
        <p class="card-text"></p>
        <div *ngIf="isShowResult">
            <div *ngFor="let user of newdata">

                <h1 style="color: blue;">{{user.district}}</h1>
            </div>
            <br>
        </div>

    </div>
</div>
<div class="card" style="width: 24rem;height:30rem;display: table-cell;text-align:center;">
    <div class="card-body">
        <h2 class="card-title">Last updated Recovered cases</h2>
        <br>
        <p class="card-text"></p>
        <div *ngIf="isShowResult">
            <div *ngFor="let user of newdata">
                <h1 style="color: rgb(2, 209, 2);">{{user.recover}}</h1>

            </div>
            <br>
        </div>

    </div>
</div>
</div>

<div class="container">
    <div style="margin-top:50px;" class="mainbox col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2">

```

```

<div class="panel panel-info" >
    <div class="panel-heading">
        <div class="panel-title text-center">TamilNadu district
corona cases</div>

    </div>

    <div style="padding-top:30px" class="panel-body" >

        <br>
        <!-- <input type="text" style="width: 100%;padding: 10px;"
placeholder="Serach the district" #country>
        <br><br>
        <button type="button" class="btn-success"
(click)="getweatherbtn(country.value)">Search</button> -->
        <!-- <input type="text" style="width: 100%;padding:
10px;" placeholder="Serach the Date" #Date>
        <br><br>
        <button type="button" class="btn-success"
(click)="getDatebtn(Date.value)">Date</button>
        -->
        <!-- <div class="search-hero">
            <input class="form-control" type="text" name="search"
[(ngModel)]="filteredString" autocomplete="off" placeholder="Start
searching for a hero by id or name or country">
            </div>

        </div>
    </div> -->
    <select name="country" style="width: 100%;padding: 10px;"
#country (change)="pull(country.value)" >
        <option value="select">Select District</option>
        <option value="all">All</option>
        <option value="Airport Quarantine">Airport
Quarantine</option>
        <option value="Ariyalur">Ariyalur</option>
        <option value="Chengalpattu">Chengalpattu</option>
        <option value="Chennai">Chennai</option>
        <option value="Coimbatore">Coimbatore</option>
        <option value="Cuddalore">Cuddalore</option>
        <option value="Dharmapuri">Dharmapuri</option>
        <option value="Dindigul">Dindigul</option>
        <option value="Erode">Erode</option>
        <option value="Kallakurichi">Kallakurichi</option>
        <option value="Kancheepuram">Kancheepuram</option>
        <option value="Kanyakumari">Kanyakumari</option>

```

```

        <option value="Karur">Karur</option>
        <option value="Krishnagiri">Krishnagiri</option>
        <option value="Madurai">Madurai</option>
        <option value="Mayiladuthurai">Mayiladuthurai</option>
        <option value="Nagapattinam">Nagapattinam</option>
        <option value="Namakkal">Namakkal</option>
        <option value="Nilgiris">Nilgiris</option>
        <option value="Other State">Other State</option>
        <option value="Perambalur">Perambalur</option>
        <option value="Pudukkottai">Pudukkottai</option>
        <option value="Railway Quarantine">Railway
Quarantine</option>
        <option value="Ramanathapuram">Ramanathapuram</option>
        <option value="Ranipet">Ranipet</option>
        <option value="Salem">Salem</option>
        <option value="Sivaganga">Sivaganga</option>
        <option value="Tenkasi">Tenkasi</option>
        <option value="Thanjavur">Thanjavur</option>
        <option value="Theni">Theni</option>
        <option value="Thiruvallur">Thiruvallur</option>
        <option value="Thiruvarur">Thiruvarur</option>
        <option value="Thoothukkudi">Thoothukkudi</option>
        <option value="Tiruchirappalli">Tiruchirappalli</option>
        <option value="Tirunelveli">Tirunelveli</option>
        <option value="Tirupathur">Tirupathur</option>
        <option value="Tiruppur">Tiruppur</option>
        <option value="Tiruvannamalai">Tiruvannamalai</option>
        <option value="Vellore">Vellore</option>
        <option value="Viluppuram">Viluppuram</option>
        <option value="Virudhunagar">Virudhunagar</option>

    </select>
</div>

<div *ngIf="isShowResult">

    <div class="row">

        <div class="col">

            <table class="table">
                <thead>
                    <tr>

```

```

        <th>Country</th>
        <th>Date</th>
        <th>Cases</th>
        <th>Recovered</th>
        <th>Decreased</th>

    </tr>
</thead>

<tbody>
    <tr *ngFor="let user of newdata">

        <td>{{user.country}}</td>
        <td>{{user.Date}}</td>
        <td>{{user.cases}}</td>
        <td>{{user.Recovered}}</td>
        <td>{{user.Deceased}}</td>

    </tr>
</tbody>
</table>
</div>
</div>

</div>
<br><br><br><br><br><br><br>
</div>

</div>

import { Component,OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ApiService } from '../api.service';
@Component({
    selector: 'app-district',
    templateUrl: './district.component.html',
    styleUrls: ['./district.component.css']
})
export class DistrictComponent implements OnInit {

    // isShowResult=false;
    // country='Ariyalur';
    // data=[]

```

```

//  constructor(public api: ApiService) { }

//  ngOnInit(): void {
//  }

//  getweatherbtn(country:string){
//    this.api.getdistrict(country).subscribe((data:any)=>{
//      this.isShowResult=true;
//      console.log(data);

//      this.data=data;
//    },(err:any)=>{
//      console.log(err);
//    },()=>{
//      console.log("Api call completed")
//    });
//  }

//  getDatebtn(Date:string){
//    this.api.getdate(Date).subscribe((data:any)=>{
//      this.isShowResult=true;
//      console.log(data);

//      this.result=data;
//    },(err:any)=>{
//      console.log(err);
//    },()=>{
//      console.log("Api call completed")
//    });
//  }

newdata:any[]=[];
isShowResult=false;
constructor(public api:ApiService){ }
  ngOnInit():void{
    this.pull("all");
  }

pull(country:string){
  this.api.getNew(country).subscribe((data:any)=>{
    console.log(data);
    this.isShowResult=true;
    this.newdata=data;
  },(err:any)=>{
    console.log(err);
  },()=>{
    console.log("completed!");
  });
}

```

```

    })
  }
}

```

EditPatient

```

<div style="background-image: url(assets/corv.jpg);background-size:
100px;background-repeat: no-repeat;background-attachment: fixed;background-
size: cover;opacity: 0.8; ">
  <h2 style="text-align:center;color: rgb(0, 0, 0);">Edit Patient's
Detail</h2>

  <div class="containe">
    <h3 style="color: rgb(195, 241, 195);">{{msg}}</h3>
    <form #reg="ngForm" *ngIf="userData" (ngSubmit)="edituser(reg)" >
      <label for="fname" style="color: black;font-size: large;"> Name :
&nbsp;</label>
      <input type="text" id="fname" name="Name" placeholder="Your
name.." pattern="^[a-zA-Z]+$" #Name="ngModel" required
[ngModel]="userData.Name"><br>
      <p class="alert alert-danger" *ngIf="Name.errors?.required &&
Name.touched">please enter Name</p>
      <label for="date" style="color: black;font-size: large;">Date of Birth
: &nbsp;</label>
      <input type="date" name="date" id="date" placeholder="dd-mm-yyyy"
required #date="ngModel" [ngModel]="userData.date"><br>
      <p class="alert alert-danger" *ngIf="date.errors?.required &&
date.touched">please enter Date of Birth</p>

      <label for="lname" style="color: black;font-size: large;">Email Id :
&nbsp;</label>
      <input type="email" id="lname" name="email" placeholder="Your email.."
required email #email="ngModel" [ngModel]="userData.email"><br>
      <p class="alert alert-danger" *ngIf="email.errors?.required &&
email.touched">please enter email Id</p>
      <p class="alert alert-danger" *ngIf="email.errors?.email &&
email.touched">please enter valid Email id</p>
      <label for="lname" style="color: black;font-size: large;">Address :
&nbsp;</label>
      <input type="text" id="lname" name="Address" placeholder="your
Address" pattern="^[a-zA-Z]+$" #country="ngModel" required
[ngModel]="userData.Address"><br>
      <p class="alert alert-danger" *ngIf="country.errors?.required &&
country.touched">please enter Address</p><br>

```



```

    });
  }

  edituser(form:NgForm){
    form.value._id=this.userData._id;
    this.users.editsingledata(form.value).subscribe((data:string)=>{
      console.log(data);
      this.msg=data
    },(error:any)=>{
      console.log(error);
      this.msg="somethign went wrong!!"
    })
  }
}

```

Home component

```

.top-left {
  position: absolute;
  top: 8px;
  left: 16px;
}

.container {
  position: relative;
  text-align: center;
  color: white;
}

.wv{
  margin-left: 12pc;
}

.footer-dark {
  padding: 50px 0;
  color: #f0f9ff;
  background-color: #282d32
}

.footer-dark h3 {

```

```

margin-top: 0;
margin-bottom: 12px;
font-weight: bold;
font-size: 16px
}

.footer-dark ul {
padding: 0;
list-style: none;
line-height: 1.6;
font-size: 14px;
margin-bottom: 0
}

.footer-dark ul a {
color: inherit;
text-decoration: none;
opacity: 0.6
}

.footer-dark ul a:hover {
opacity: 0.8
}

@media (max-width:767px) {
.footer-dark .item:not(.social) {
text-align: center;
padding-bottom: 20px
}
}

.footer-dark .item.text {
margin-bottom: 36px
}

@media (max-width:767px) {
.footer-dark .item.text {
margin-bottom: 0
}
}

.footer-dark .item.text p {
opacity: 0.6;
margin-bottom: 0
}

.footer-dark .item.social {
text-align: center

```

```

}

@media (max-width:991px) {
  .footer-dark .item.social {
    text-align: center;
    margin-top: 20px
  }
}

.footer-dark .item.social>a {
  font-size: 20px;
  width: 36px;
  height: 36px;
  line-height: 36px;
  display: inline-block;
  text-align: center;
  border-radius: 50%;
  box-shadow: 0 0 0 1px rgba(255, 255, 255, 0.4);
  margin: 0 8px;
  color: #fff;
  opacity: 0.75
}

.footer-dark .item.social>a:hover {
  opacity: 0.9
}

.footer-dark .copyright {
  text-align: center;
  padding-top: 24px;
  opacity: 0.3;
  font-size: 13px;
  margin-bottom: 0
}
<div class="container-fluid">

  <nav class="navbar navbar-expand-lg
    navbar-dark justify-content-between text-white"
    style="background-color: #0a1a27;">
    <a class="navbar-brand" href="#" >

      <h4> COV19 Tracker</h4>
    </a>
    <button class="navbar-toggler"
      type="button"
      data-toggle="collapse"
      data-target="#navbarNavDropdown01"
      aria-controls="navbarNavDropdown01"

```

```

        aria-expanded="false"
        aria-label="Toggle navigation"
        style="outline-color:rgb(31, 80, 187);background-color:
blue;">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse m-2 p-4 navbar-nav text-right"
id="navbarNavDropdown01" style="background-color: rgb(179,
179, 235)">

        <ul class="navbar-nav text-right ml-4 navbar-nav pull-right text-
dark" >

            <li class="nav-item " *ngIf="!user.isLoggedIn()">
                <a class="nav-link text-dark" routerLink="/login"
target="_blank" style="font-size: 1.4em;">
                    Login &nbsp;  
                </a>
            </li>

            <!-- inactive link text item of menu -->
            <li class="nav-item">
                <a class="nav-link text-dark" routerLink="/News"
style="font-size: 1.4em;">News&nbsp;  </a>
            </li>

            <!-- dropdown item of menu -->
            <!-- <li class="nav-item dropdown">
                <a class="nav-link text-dark" routerLink="/patient's
details" style="font-size: 1.4em;">
                    Patient's details&nbsp;  
                </a>

            </li>
            -->

            <li class="nav-item">
                <a routerLink="/district" class="nav-link text-
dark"style="font-size: 1.4em; ">Covid Cases&nbsp;  </a>

            </li>
            <li class="nav-item">

```

```

        <a routerLink="/adduseradmin" class="nav-link text-
dark"style="font-size: 1.4em;">patient detail&nbsp;</a>
    </li>
    <li class="nav-item">
        <a routerLink="/contact" class="nav-link text-
dark"style="font-size: 1.4em;">Contact&nbsp;</a>
    </li>
    <li class="nav-item" *ngIf="user.isLoggedIn()">
        <a routerLink="/logout" class="nav-link text-
dark"style="font-size: 1.4em;" (click)="doLogout()">logout&nbsp;</a>
    </li>
</ul>
</div>
</nav>
</div>

<div style=" background-color: rgb(192, 183, 247);" ><br><br>
    <div class="ww">
        <h2>Co-Win</h2>
        <p>Winning over covid-19</p>
    </div>
    <br>
<div class="container">

    <div id="myCarousel" class="carousel slide" data-ride="carousel">
        <!-- Indicators -->
        <ol class="carousel-indicators">
            <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
            <li data-target="#myCarousel" data-slide-to="1"></li>
            <li data-target="#myCarousel" data-slide-to="2"></li>
            <li data-target="#myCarousel" data-slide-to="3"></li>

        </ol>

        <!-- Wrapper for slides -->
        <div class="carousel-inner">
            <div class="item active">
                
                <div class="top-left"style="font-size:
1.5em;"><br><br><br><br><br><br><br><br><br>"India is set to defeat<br>
                Covid-19. Every Indian is making it possible."<br>
                - Narendra Modi</div>
            </div>
            <div class="item">
                

```

```

</div>

<div class="item">
  
</div>

<div class="item">
  

</div>

</div>

<!-- Left and right controls -->
<a class="left carousel-control" href="#myCarousel" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-left"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel" data-slide="next">
  <span class="glyphicon glyphicon-chevron-right"></span>
  <span class="sr-only">Next</span>
</a>
</div>
</div>

<br><br>
<div _ngcontent-kym-c117="" class="easyVaccination">

<div _ngcontent-kym-c117="" class="container">
<h1 _ngcontent-kym-c117="" class="text-center accessibility-plugin-ac text-dark">Getting Vaccinated Is As Easier Than Ever</h1>
<div _ngcontent-kym-c117="" class="row mt-5">

<div _ngcontent-kym-c117="" class="col-sm-12 col-md-4 mb-5 text-center">
<p _ngcontent-kym-c117="" class="accessibility-plugin-ac">
  <span _ngcontent-kym-c117="" class="steps-count dashed-line text-dark">1 &nbsp;</span><span _ngcontent-kym-c117="" class="text-dark">Book an Appointment On <br>CoWIN or Walk Into Any <br>Vaccination Center</span></p><a _ngcontent-kym-c117="" title="How to book your appointment on CoWIN" aria-

```

```

label="How to book your appointment on CoWIN" target="_blank"
href="https://prod-cdn.preprod.co-
vin.in/assets/pdf/User_Guide_Citizen%20registration_18%2B.pdf"
class="accessibility-plugin-ac">How to book your appointment on
CoWIN</a></div><div _ngcontent-kym-c117="" class="col-sm-12 col-md-4 mb-5
text-center"><p _ngcontent-kym-c117="" class="accessibility-plugin-ac"><span
_ngcontent-kym-c117="" class="steps-count dashed-line text-
dark">2 &nbsp;  </span><span _ngcontent-kym-c117="" class="text-dark">Get Your
Vaccination <br> Safely at the Time of Your <br> Appointment</span></p><a
_ngcontent-kym-c117="" title="Do's and don'ts for getting vaccinated" aria-
label="Do's and don'ts for getting vaccinated" target="_blank"
class="accessibility-plugin-ac" href="https://prod-cdn.preprod.co-
vin.in/assets/pdf/Dos_and_Donts_for_Citizens.pdf">Dos and dont's for getting
vaccinated</a></div><div _ngcontent-kym-c117="" class="col-sm-12 col-md-4 mb-5
text-center"><p _ngcontent-kym-c117="" class="accessibility-plugin-ac"><span
_ngcontent-kym-c117="" class="steps-count text-dark">3 &nbsp;  </span><span
_ngcontent-kym-c117="" class="text-dark">Download Your Vaccination <br>
Certificate from CoWIN <br> and Wait for Dose #2</span></p><a
_ngcontent-kym-c117="" title="Download your vaccine certificate"
target="_blank" aria-label="Download your vaccine certificate"
class="accessibility-plugin-ac"
href="https://selfregistration.cowin.gov.in/vaccination-certificate">Download
Your Vaccine Certificate</a></div></div></div></div></div>

```

```

<div class="footer-dark">
  <footer>
    <div class="container">
      <div class="row">
        <div class="col-md-3 item">
          <h3>Donation</h3>
          <ul>
            <li><a href="#">plasma</a></li>
            <li><a href="#">blood</a></li>
            <li><a href="#">organ</a></li>
          </ul>
        </div>
        <div class="col-md-3 item">
          <h3>Services</h3>
          <ul>
            <li><a href="#">Download cerificate </a></li>
            <li><a href="#">verification</a></li>
            <li><a href="#">Hosting</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
        <div class="col-md-3 item">
            <h3>About</h3>
            <ul>
                <li><a href="#">government</a></li>
                <li><a href="#">vaccination</a></li>
                <li><a href="#">Instruction</a></li>
            </ul>
        </div>
        <div class="col-md-3 item text">
            <h3>cowin.com</h3>
            <p>vaccinate if you are not vaccinated</p>
        </div>

        <div class="col item social"><a href="#"><i class="fa fa-
facebook"></i></a><a href="#"><i class="fa fa-twitter"></i></a><a href="#"><i
class="fa fa-youtube"></i></a><a href="#"><i class="fa fa-
instagram"></i></a><a href="#"><i class="fa fa-google"></i></a></div>
    </div>
    <p class="copyright">cowin.com © 2021</p>
</div>
</footer>
</div>

```

```

import { Component, OnInit } from '@angular/core';
import { ContactService } from '../contact.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  constructor(public user : ContactService,public myRouter : Router) { }

  ngOnInit(): void {
  }
  doLogout(){
    localStorage.clear();
    this.myRouter.navigateByUrl("/");
  }
}

```

Login component

```
.login-box{
  width: 320px;
  height: 420px;
  background: black;
  color: rgb(236, 206, 206);
  top: 50%;
  left: 50%;
  position: absolute;
  transform: translate(-50%,-50%);
  box-sizing: border-box;
  padding: 70px 30px;
}

.avatar{
  width: 100px;
  height: 100px;
  border-radius: 50%;
  position: absolute;
  top: -50px;
  left: calc(50% - 50px);
}

h1{
  margin: 0;
  padding: 0 0 20px;
  text-align: center;
  font-size: 22px;
}

.login-box p{
  margin: 0;
  padding: 0;
  font-weight: bold;
}

.login-box input{
  width: 100%;
  margin-bottom: 20px;
}

.login-box input[type="email"], input[type="password"]
{
  border: none;
  border-bottom: 1px solid #fff;
  background: transparent;
  outline: none;
  height: 40px;
  color: rgb(245, 195, 195);
  font-size: 16px;
}
```

```

}
.login-box input[type="submit"]
{
    border: none;
    outline: none;
    height: 40px;
    background: #1c8adb;
    color: #fff;
    font-size: 18px;
    border-radius: 20px;
}
.login-box input[type="submit"]:hover
{
    cursor: pointer;
    background: #39dc79;
    color: rgb(223, 204, 204);
}

.login-box a{
    text-decoration: none;
    font-size: 14px;
    color: rgb(230, 220, 220);
}
.login-box a:hover
{
    color: #39dc79;
}

```

```

<div >
    <img [src]="myimage" style="width: 100%;height: 100%;background-size:
cover;background-attachment: fixed;">
</div>

    <div class="login-box">
        <p style="color: green;">

```



```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';

import { ContactService } from '../contact.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  myimage:string= "assets/blog.jpg";
  msg : string;

  constructor(public user: ContactService, public myRouter : Router) { }

  ngOnInit(): void {
  }

  doLogin(form:NgForm)
  {
    console.log(form.value);

    this.user.userLogin(form.value).subscribe((data:any[])=>{

```

```

        console.log(data);

        if(data.length==0){
            this.msg="Invalid";
        }else{
            localStorage.setItem("loggeduser",data[0]._id);
            this.myRouter.navigateByUrl("/");
        }
        form.reset();
    },(error:any)=>{
        console.log(error);
        this.msg="something went wrong!!";
    });
}
}

```

loginAmin

```

.login-box{
    width: 320px;
    height: 420px;
    background: black;
    color: rgb(236, 206, 206);
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%,-50%);
    box-sizing: border-box;
    padding: 70px 30px;
}
.avatar{
    width: 100px;
    height: 100px;
    border-radius: 50%;
    position: absolute;
    top: -50px;
    left: calc(50% - 50px);
}
h1{
    margin: 0;
    padding: 0 0 20px;
}

```

```

    text-align: center;
    font-size: 22px;
}
.login-box p{
    margin: 0;
    padding: 0;
    font-weight: bold;
}
.login-box input{
    width: 100%;
    margin-bottom: 20px;
}
.login-box input[type="email"], input[type="password"]
{
    border: none;
    border-bottom: 1px solid #fff;
    background: transparent;
    outline: none;
    height: 40px;
    color: rgb(245, 195, 195);
    font-size: 16px;
}
.login-box input[type="submit"]
{
    border: none;
    outline: none;
    height: 40px;
    background: #1c8adb;
    color: #fff;
    font-size: 18px;
    border-radius: 20px;
}
.login-box input[type="submit"]:hover
{
    cursor: pointer;
    background: #39dc79;
    color: rgb(223, 204, 204);
}

.login-box a{
    text-decoration: none;
    font-size: 14px;
    color: rgb(230, 220, 220);
}
.login-box a:hover
{
    color: #39dc79;
}

```

```

<div class="login-box" style="background-color:rgb(1, 1, 2)">
  <p style="color: green;">
    {{msg}}
  </p><br>
<form #f="ngForm" (ngSubmit)="doLoginadmin(f)">
  
  <h1>Admin</h1>

  <p>Useremail</p>
  <input type="email" name="useremail" placeholder="Enter useremail"
ngModel>
  <p>Password</p>
  <input type="password" name="password" placeholder="Enter
Password" ngModel>
  <input type="submit" name="submit" value="Login">

</form>
</div>

```

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';
import { ContactService } from '../contact.service';

@Component({
  selector: 'app-loginadmin',
  templateUrl: './loginadmin.component.html',
  styleUrls: ['./loginadmin.component.css']
})
export class LoginadminComponent implements OnInit {

  msg : string;

  constructor(public user: ContactService, public myRouter : Router) { }

  ngOnInit(): void {
  }

  doLoginadmin(form:NgForm)
  {
    console.log(form.value);

    this.user.userLoginadmin(form.value).subscribe((data:any[])=>{

```

```

        console.log(data);

        if(data.length==0){
            this.msg="Invalid";
        }else{
            localStorage.setItem("loggeduser",data[0]._id);
            this.myRouter.navigateByUrl("/adduser");
        }
        form.reset();
    },(error:any)=>{
        console.log(error);
        this.msg="something went wrong!!";
    });
}
}

```

News component

```

.containe{
    background-color: rgb(176, 176, 247);
}
.top-left {
    position: absolute;
    top: 8px;
    left: 16px;
}

.container {
    position: relative;
    text-align: center;
    color: white;
}

.wv{
    margin-left: 12pc;
}

<div class="container-fluid">

    <nav class="navbar navbar-expand-lg
        navbar-dark justify-content-between text-white"
        style="background-color: #0a1a27;">
        <a class="navbar-brand" href="#" >

```

```

    <h4> COV19 Tracker</h4>
  </a>
  <button class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navbarNavDropdown01"
    aria-controls="navbarNavDropdown01"
    aria-expanded="false"
    aria-label="Toggle navigation"
    style="outline-color:rgb(31, 80, 187);background-color:
blue;">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse m-2 p-4 navbar-nav text-right"
    id="navbarNavDropdown01" style="background-color: rgb(179,
179, 235)">

    <ul class="navbar-nav text-right ml-4 navbar-nav pull-right text-
dark" >

      <!-- inactive link text item of menu -->
      <!-- <li class="nav-item">
        <a class="nav-link text-dark" routerLink="/News"
style="font-size: 1.4em;">News&nbsp;</a>
      </li> -->

      <!-- dropdown item of menu -->
      <!-- <li class="nav-item dropdown">
        <a class="nav-link text-dark" routerLink="/patient's
details" style="font-size: 1.4em;">
          Patient's details&nbsp;</a>
      </li>
      -->

      <li class="nav-item">
        <a routerLink="/district" class="nav-link text-
dark"style="font-size: 1.4em; ">Covid Cases&nbsp;</a>

```

```

        </li>
        <li class="nav-item">
            <a routerLink="/adduseradmin" class="nav-link text-
dark"style="font-size: 1.4em;">patient detail&nbsp;</a>
        </li>
        <li class="nav-item">
            <a routerLink="/contact" class="nav-link text-
dark"style="font-size: 1.4em;">Contact&nbsp;</a>
        </li>
    </ul>
</div>
</nav>
</div>

<div class="containe">
<div class="container">
    <div style="margin-top:50px;" class="mainbox col-md-6 col-md-offset-3 col-
sm-8 col-sm-offset-2">
        <div class="panel panel-primary" >
            <div class="panel-heading">
                <div class="panel-title text-center bg-info text-
dark"><h2>News About Covid19</h2></div>
            </div>
        </div>
    </div>
</div>
<div style="padding-top:30px" class="panel-body">
<select name="newscat" style="width: 150px;padding: 10px;" #newscat
(change)="pull(newscat.value)">
    <option value="select">Select Category</option>
    <option value="all">All</option>
</select>
</div>
<div *ngIf="!isShowresult" style="text-align: center;">
    Loading..
</div>

<div class="container" *ngIf="isShowresult">
<div *ngFor="let news of newsdata">
<div class="col-md-3 newstab">



<p style="font-weight: 900;">{{news.title | slice:0:50}}...</p>
<p>{{news.discription | slice:0:50}}...</p>

```

```

<a href="{{news.url}}" target="_blank"><button type="button" class="btn btn-
outline-danger">Read More... </button> </a> <hr size="5"
color="black">
<br>

</div>
</div>

</div>
</div>

import { Component, OnInit } from '@angular/core';
import { ApiService } from '../api.service';

@Component({
  selector: 'app-news',
  templateUrl: './news.component.html',
  styleUrls: ['./news.component.css']
})
export class NewsComponent implements OnInit {
  isShowresult=false;
  newsdata:any[]=[];
  constructor(public api:ApiService) { }

  ngOnInit(): void {
    this.pull("all");
  }

  pull(newsCat:string){
    this.api.getNews(newsCat).subscribe((data:any)=>{
      console.log(data);
      this.isShowresult=true;
      this.newsdata=data.articles;
    },(err:any)=>{
      console.log(err);
    },()=>{
      console.log("completed");
    })
  }
}

```

Patient component

```

input[type=text] {
  width:40%;
  padding: 12px;
  border: 1px solid #ccc;
}

```

```

border-radius: 4px;
box-sizing: border-box;
margin-top: 6px;
margin-bottom: 16px;

}
input[type=email]{
width:40%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
margin-top: 6px;
margin-bottom: 16px;
}
input[type=date]{
width:40%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
margin-top: 6px;
margin-bottom: 16px;
}
form{
border-radius:5px;

padding: 20px;

}
input.ng-invalid{
border-color:red;
padding: 20px;
}
input.ng-untouched{
border-color:green;
}

button[type=submit] {
background-color: #04AA6D;
color: white;
padding: 12px 20px;
border: none;
border-radius: 4px;
cursor: pointer;
}

```

```

button[type=submit]:hover {
  background-color: #45a049;
}

.containe {
  border-radius: 5px;
  text-align: center;
  padding: 20px;
}

.footer-dark {
  padding: 50px 0;
  color: #f0f9ff;
  background-color: #282d32
}

.footer-dark h3 {
  margin-top: 0;
  margin-bottom: 12px;
  font-weight: bold;
  font-size: 16px
}

.footer-dark ul {
  padding: 0;
  list-style: none;
  line-height: 1.6;
  font-size: 14px;
  margin-bottom: 0
}

.footer-dark ul a {
  color: inherit;
  text-decoration: none;
  opacity: 0.6
}

.footer-dark ul a:hover {
  opacity: 0.8
}

@media (max-width:767px) {
  .footer-dark .item:not(.social) {
    text-align: center;
    padding-bottom: 20px
  }
}

```

```

.footer-dark .item.text {
    margin-bottom: 36px
}

@media (max-width:767px) {
    .footer-dark .item.text {
        margin-bottom: 0
    }
}

.footer-dark .item.text p {
    opacity: 0.6;
    margin-bottom: 0
}

.footer-dark .item.social {
    text-align: center
}

@media (max-width:991px) {
    .footer-dark .item.social {
        text-align: center;
        margin-top: 20px
    }
}

.footer-dark .item.social>a {
    font-size: 20px;
    width: 36px;
    height: 36px;
    line-height: 36px;
    display: inline-block;
    text-align: center;
    border-radius: 50%;
    box-shadow: 0 0 0 1px rgba(255, 255, 255, 0.4);
    margin: 0 8px;
    color: #fff;
    opacity: 0.75
}

.footer-dark .item.social>a:hover {
    opacity: 0.9
}

.footer-dark .copyright {
    text-align: center;
    padding-top: 24px;
}

```



```

        </div>

        <div class="col item social"><a href="#"><i class="fa fa-
facebook"></i></a><a href="#"><i class="fa fa-twitter"></i></a><a href="#"><i
class="fa fa-youtube"></i></a><a href="#"><i class="fa fa-
instagram"></i></a><a href="#"><i class="fa fa-google"></i></a></div>
    </div>
    <p class="copyright">cowin.com © 2021</p>
</div>
</footer>
</div>

```

```

import { Component, OnInit } from '@angular/core';
import { ContactService } from '../contact.service';
import { NgForm } from '@angular/forms';

@Component({
  selector: 'app-patient',
  templateUrl: './patient.component.html',
  styleUrls: ['./patient.component.css']
})
export class PatientComponent implements OnInit {
  msg : string;
  usernameAvail=false;
  constructor(public user:ContactService) { }

  ngOnInit(): void {
  }

  Patient(reg:NgForm){

    console.log(reg.value);

    this.user.patientcon(reg.value).subscribe((data:string)=>{

      console.log(data);

      this.msg = data;

      reg.reset();

    }, (error:any)=>{

      console.log(error);

      this.msg = "Something Went Wrong";
    })
  }
}

```

```

    });

    }

    unamecheck(Name:string){
this.user.usernameAvail(Name).subscribe((data:any[])=>{
console.log(data);
if(data.length==0){
    this.msg="username Available!!";
    this.usernameAvail=true;
}else{
    this.msg="username taken!";
    this.usernameAvail=false;
}
}, (error:any)=>{
    console.log(error);
});
    }

}

```

Register component

```

<div style="background-color: rgb(255, 249, 240);">
<div class="container">
    <div style="margin-top:100px;" class="mainbox col-md-6 col-md-offset-3
col-sm-8 col-sm-offset-2">
        <div class="panel panel-info" >
            <div class="panel-heading">
                <div class="panel-title bg-white text-dark"style="text-
align:center">Register</div>
            </div>
            <div style="padding-top:30px" class="panel-body" >
                <h3 style="color: green;">
                    {{msg}}
                </h3>
                <form #f="ngForm"
                (ngSubmit)="Regis(f)">
                    <div style="margin-bottom: 25px" class="input-group">
                        <!-- <span class="input-group-addon"><i
class="glyphicon glyphicon-user"></i></span> -->

```

```

        <input type="text" class="form-control" name="username" ngModel
#username="ngModel" placeholder="Username" required
minlength="6">

        </div>

        <p class="alert alert-danger"
*ngIf="username.errors?.required &&
username.touched">

            Please Enter Username
        </p>
        <p class="alert alert-danger"
*ngIf="username.errors?.minlength &&
username.touched">

            Please Enter Minimum 6 Characters
        </p>
        <div style="margin-bottom: 25px" class="input-group">
            <!-- <span class="input-group-addon"><i
class="glyphicon glyphicon-envelope"></i></span> -->
            <input type="text" class="form-control" name="useremail" ngModel
#useremail="ngModel" placeholder="Email" required
email>

        </div>

        <p class="alert alert-danger"
*ngIf="useremail.errors?.required &&
useremail.touched">

            Please Enter Your Email
        </p>
        <p class="alert alert-danger"
*ngIf="useremail.errors?.email &&
useremail.touched">

            Please Enter Valid Email
        </p>
        <div style="margin-bottom: 25px" class="input-group">
            <!-- <span class="input-group-addon"><i
class="glyphicon glyphicon-lock"></i></span> -->
            <input type="password" class="form-control" name="password"
ngModel #password="ngModel" placeholder="Password" required>
        </div>

        <div style="margin-bottom: 25px" class="input-group">
            <!-- <span class="input-group-addon"><i
class="glyphicon glyphicon-phone"></i></span> -->
            <input type="text" class="form-control" name="uphone"
ngModel placeholder="Phone Number" #userphone="ngModel" pattern="^(0|[1-9][0-
9]*)$">

        </div>

```

```

        <p class="alert alert-danger"
        *ngIf="userphone.errors?.pattern">
            Please Enter Nos Only
        </p>
        <div style="margin-top:10px" class="form-group">
            <!-- Button -->
            <div class="col-sm-12">
                <input type="submit" class="btn btn-success" value="Register"
[disabled]="f.invalid" /> &nbsp;
                Already have Account? <a
routerLink="/login">Login</a>
            </div>
        </div>
    </form>
    <br> <br><br>

    </div>
</div>
</div>
</div>
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { ContactService } from '../contact.service';

@Component({
    selector: 'app-register',
    templateUrl: './register.component.html',
    styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {
    msg:string;
    constructor(public user:ContactService) { }

    ngOnInit(): void {
    }
    Regis(f:NgForm){
        console.log(f.value);

        this.user.register(f.value).subscribe((data:string)=>{

            this.msg=data;
            f.reset();
        },(error:any)=>{
            console.log(error);
            this.msg="Something Went Wrong";

```

```

    });
  }
}

```

Api service

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
@Injectable({
  providedIn: 'root'
})
export class ApiService {

  constructor(public http:HttpClient) {
    console.log("service created!!");
  }

  getNews(newsCat:string){
    return
this.http.get<any>("https://newsapi.org/v2/everything?q=covid19&sortBy=publish
edAt&apiKey=4a52fc9322914bcba8ed6859a1d8c342");
  }

  // getdistrict(country:string){

  //   return
this.http.get<any>("https://raw.githubusercontent.com/Harikrishna-
10/district5/main/Ariyalur.json");
  // }
  // getDate(Date:string){
  //   return
this.http.get<any>("https://raw.githubusercontent.com/Harikrishna-
10/district5/main/"+Date+".json")
  // }

  getNew(country:string){
    console.log(country);

    if(country=="all")
    {
      return
this.http.get<any>("https://data.covid19india.org/state_district_wise.json");
    }
    else{

```

```

        return
this.http.get<any>("https://raw.githubusercontent.com/Harikrishna-10/district5/main/"+country+".json")
    }
}

}

```

App-routing

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AdduserComponent } from './adduser/adduser.component';
import { AdduseradminComponent } from './adduseradmin/adduseradmin.component';
import { AuthGuard } from './auth.guard';
import { ContactComponent } from './contact/contact.component';
import { DistrictComponent } from './district/district.component';
import { EditpatientComponent } from './editpatient/editpatient.component';
import { HomeComponent } from './home/home.component';
import { LoginComponent } from './login/login.component';
import { LoginadminComponent } from './loginadmin/loginadmin.component';
import { LogoutComponent } from './logout/logout.component';
import { NewsComponent } from './news/news.component';
import { PatientComponent } from './patient/patient.component';
import { RegisterComponent } from './register/register.component';

const routes: Routes = [
  {path:"",component:HomeComponent},
  {path:"login",component:LoginComponent},
  {path:"register",component:RegisterComponent},
  {path:"News",component:NewsComponent,canActivate:[AuthGuard]},
  {path:"patient's details" ,
component:PatientComponent,canActivate:[AuthGuard]},
  {path:"district",component:DistrictComponent,canActivate:[AuthGuard]},
  {path:"contact",component:ContactComponent,canActivate:[AuthGuard]},
  {path:"logout",component:LogoutComponent},
  {path:"adduser",component:AdduserComponent,canActivate:[AuthGuard]},
  {path:"editpatient/:userid",component:EditpatientComponent},
  {path:"adduseradmin",component:AdduseradminComponent,canActivate:[AuthGuard]},
},
  {path:"loginadmin",component:LoginadminComponent,canActivate:[AuthGuard]}
];

@NgModule({
  imports: [RouterModule.forRoot(routes,{useHash:true})],

```

```

    exports: [RouterModule]
  })
  export class AppRoutingModule { }

```

app-component.html

```
<router-outlet></router-outlet>
```

App -gurad

```

import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';
import { ContactService } from './contact.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(public user:ContactService,public myRouter:Router){

  }

  canActivate(): boolean {

    if(!this.user.isLoggedIn()){
      this.myRouter.navigateByUrl("/");
    }
    return this.user.isLoggedIn();
  }

}

```

Contact service

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class ContactService {

  constructor(public http:HttpClient) { }

  Registerscon(data:any){
    return this.http.post<string>("http://localhost:3000/contact",data);
  }
  patientcon(data:any){
    return this.http.post<string>("http://localhost:3000/patient",data);
  }
}

```

```

userLogin(data:any){
    return this.http.post<any[]>("http://localhost:3000/login",data);
}
register(data:any){
    return this.http.post<string>("http://localhost:3000/register",data);
}
isLoggedIn(){
    return !!localStorage.getItem("loggeduser");
}
getpatient(){
    return this.http.get<any[]>("http://localhost:3000/adduser");
}
usernameAvail(Name:string){
    return this.http.get<any[]>("http://localhost:3000/unamecheck/"+Name);
}
getsingleuserdata(userid:string){
    return this.http.get<any[]>("http://localhost:3000/getuser/"+userid);
}
editsingledata(data:any){
    return this.http.put<string>("http://localhost:3000/updateuser/",data);
}
deleteuserdata(userid:number){
    return this.http.delete<string>("http://localhost:3000/deleteuser/"+userid);
}
searchuser(search:string){
    return this.http.get<any[]>("http://localhost:3000/search/"+search);
}
getuse(){
    return this.http.get<any[]>("http://localhost:3000/adduseradmin");
}
searchuse(searc:string){
    return this.http.get<any[]>("http://localhost:3000/searc/"+searc);
}
userLoginadmin(data:any){
    return this.http.post<any[]>("http://localhost:3000/loginadmin",data);
}
}

```

OUTPUT SNAPSHOT:

cloud.mongodb.com/v2/61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find

Project 0 Atlas Realm Charts

DEPLOYMENT

Databases

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

DATABASES: 1 COLLECTIONS: 4

+ Create Database

NAMESPACES

covid

login

loginadmin

patient

patient detail

covid.login

COLLECTION SIZE: 436B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: "value" } OPTIONS Apply Reset

QUERY RESULTS 1-4 OF 4

```
{ "_id": "61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find", "password": "1234", "username": "harsh" }, { "_id": "61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find", "password": "1234", "username": "harsh" }, { "_id": "61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find", "password": "1234", "username": "harsh" }, { "_id": "61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find", "password": "1234", "username": "harsh" }
```

https://cloud.mongodb.com/v2/61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/login/find

cloud.mongodb.com/v2/61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/loginadmin/find

Project 0 Atlas Realm Charts

DEPLOYMENT

Databases

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

DATABASES: 1 COLLECTIONS: 4

+ Create Database

NAMESPACES

covid

login

loginadmin

patient

patient detail

covid.loginadmin

COLLECTION SIZE: 79B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 24KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: "value" } OPTIONS Apply Reset

QUERY RESULTS 1-1 OF 1

```
{ "_id": "61854d4a2cd8b24aa870906#metrics/replicaSet/618549b3ee617015bba3df/explorer/covid/loginadmin/find", "password": "1234", "username": "harsh" }
```

System Status: All Good
©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

cloud.mongodb.com/v2/61854d4a2cdd8b24aa370906f#metrics/replicaSet/61854f9b3ee617015bba3df/explorer/covid/patient/find

Atlas

Project 0

DEPLOYMENT

Databases

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

Namespaces

covid

login

loginadmin

patient

patient detail

covid.patient

COLLECTION SIZE: 1KB TOTAL DOCUMENTS: 10 INDEXES TOTAL SIZE: 36KB

Find Indexes Scheme Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

Filter { field: "value" }

OPTIONS Apply Reset

QUERY RESULTS 1-10 OF 10

{	_id: "6036129455864"	name: "hari"	email: "har@gmail.com"	address: "chennai"	subject: "ghost"	}
{	_id: "6036129542125"	name: "hari"	email: "har@gmail.com"	address: "chennai"	subject: "ghost"	}
{	_id: "6036129579672"	name: "hari"	date: "2021-11-18"	email: "har@gmail.com"	address: "chennai"	doze: "doze1"
{	_id: "6036129579139"	name: "hari"	email: "har@gmail.com"	address: "chennai"	subject: "ghost"	}

cloud.mongodb.com/v2/61854d4a2cdd8b24aa370906f#metrics/replicaSet/61854f9b3ee617015bba3df/explorer/covid/patient%20detail/find

Atlas

Project 0

DEPLOYMENT

Databases

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

Namespaces

covid

login

loginadmin

patient

patient detail

covid.patient detail

COLLECTION SIZE: 1.5KB TOTAL DOCUMENTS: 13 INDEXES TOTAL SIZE: 36KB

Find Indexes Scheme Anti-Patterns Aggregation Search Indexes

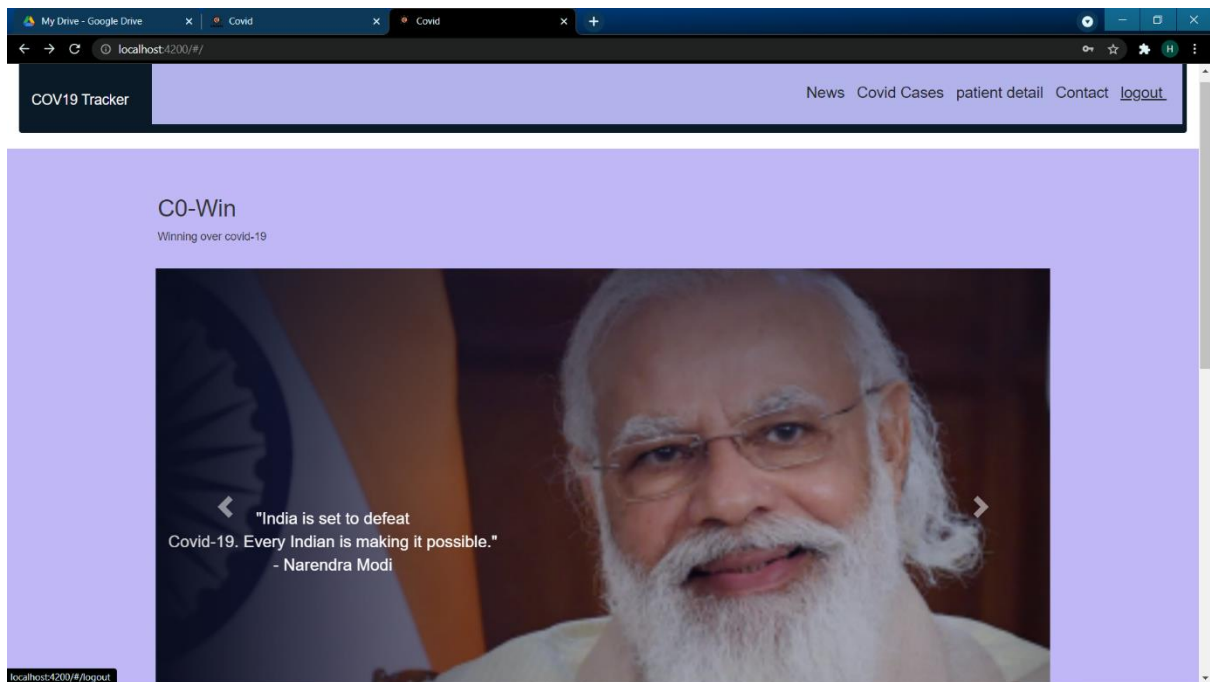
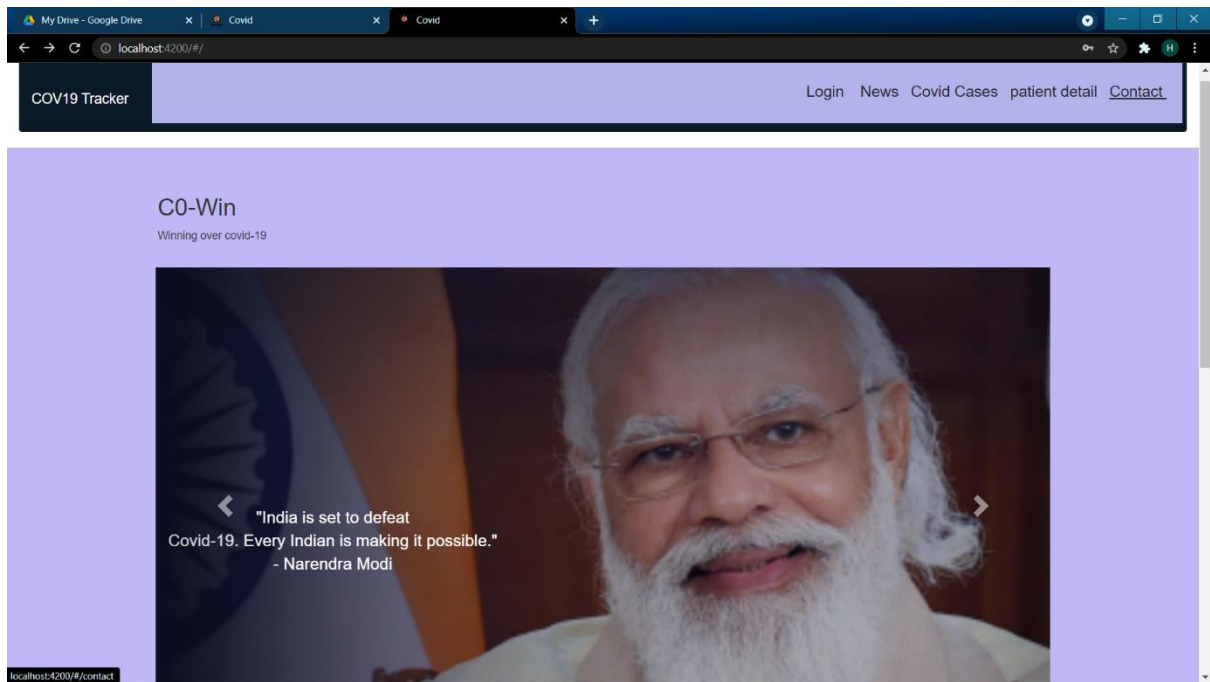
INSERT DOCUMENT

Filter { field: "value" }

OPTIONS Apply Reset

QUERY RESULTS 1-13 OF 13

{	_id: "6036129488154"	name: "hari"	date: "2021-11-18"	email: "har@gmail.com"	address: "chennai"	doze: "doze1"	}
{	_id: "6036129579672"	name: "hari"	date: "2021-11-18"	email: "har@gmail.com"	address: "chennai"	doze: "doze1"	}
{	_id: "6036129579139"	name: "hari"	date: "2021-11-18"	email: "har@gmail.com"	address: "chennai"	doze: "doze1"	}
{	_id: "60361295794955"	name: "hari"	date: "2021-11-18"	email: "har@gmail.com"	address: "chennai"	doze: "doze1"	}



My Drive - Google Drive x Covid x Covid x +

localhost:4200/#/contact

Contact Us

User Resgitered successfully

Name :

Email Id :

Address :

Subject :

My Drive - Google Drive x Covid x Covid x +

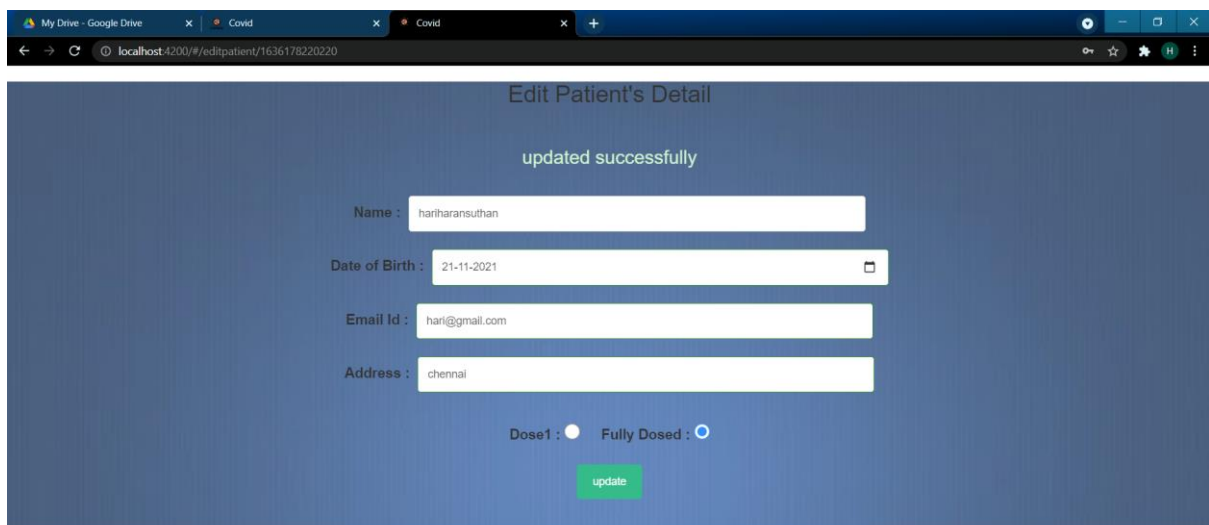
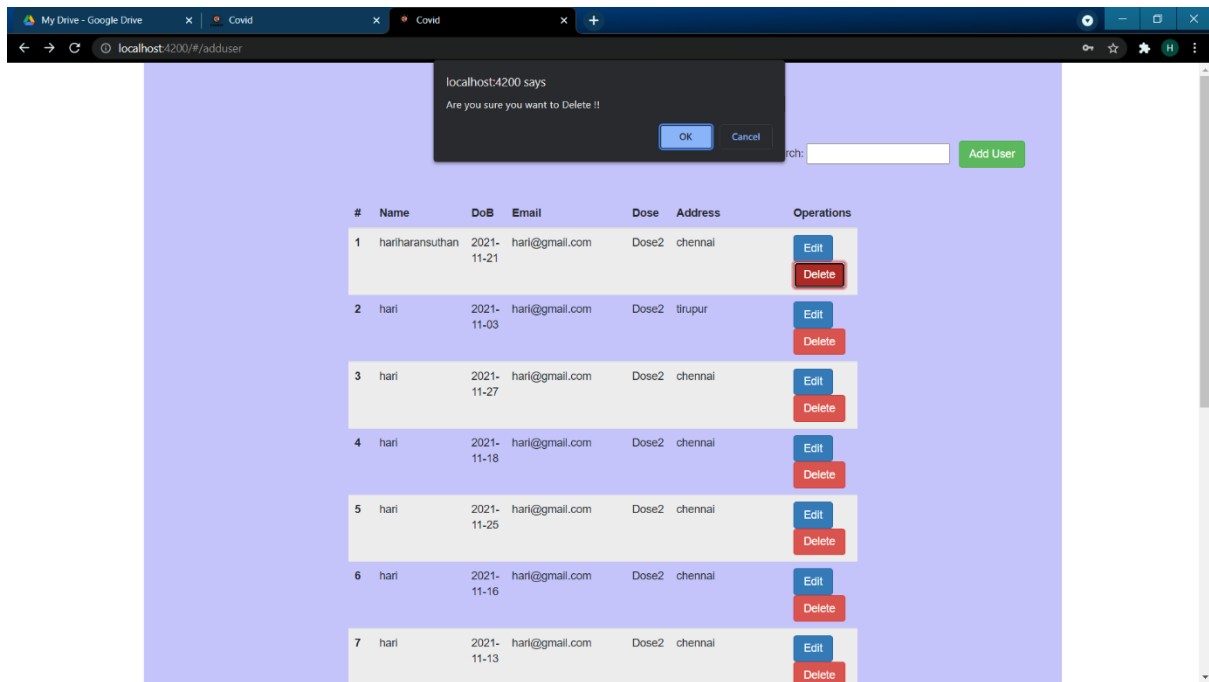
localhost:4200/#/adduser

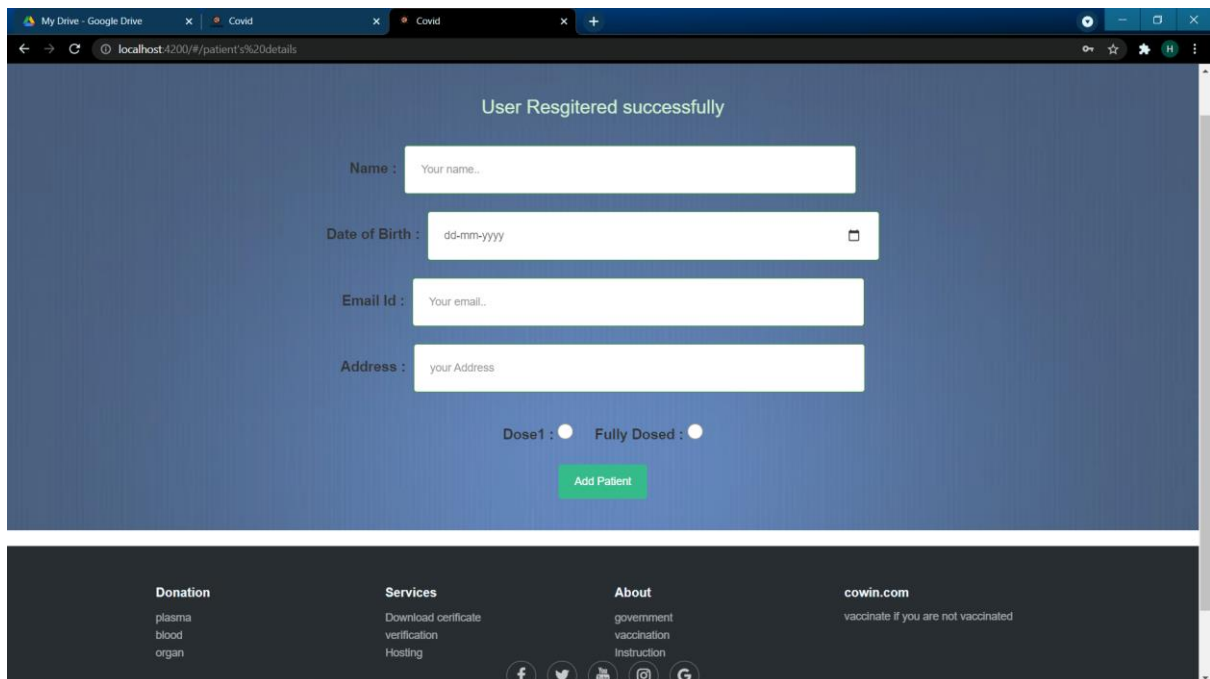
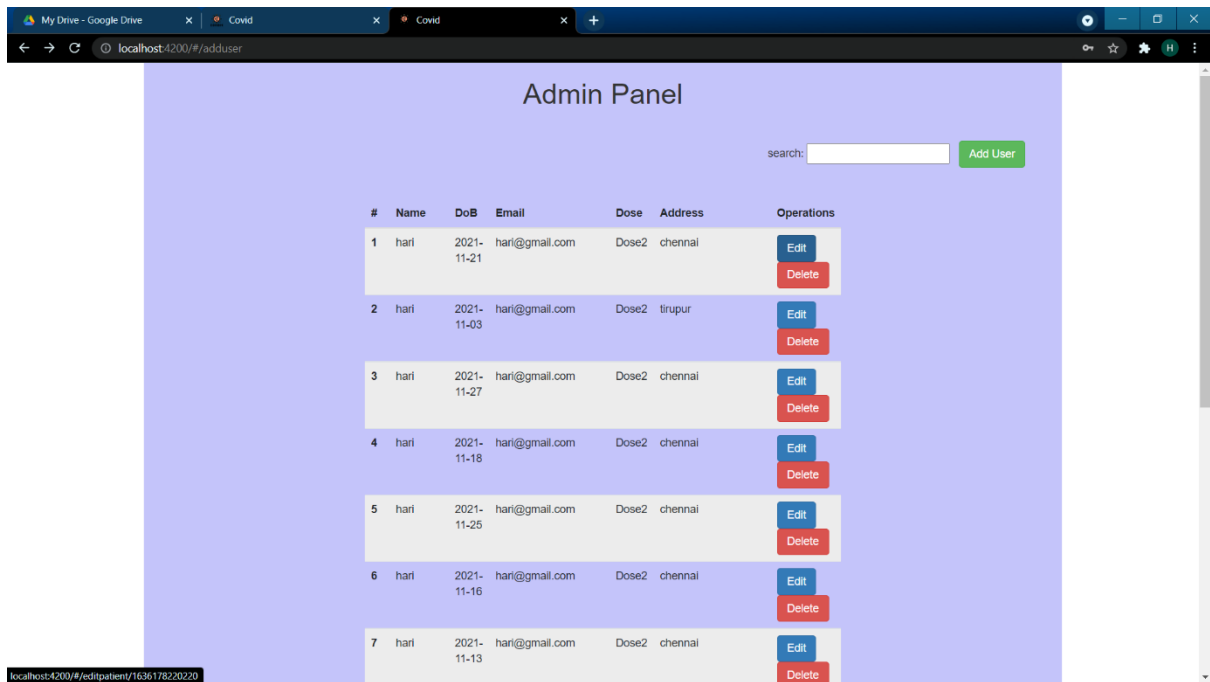
Admin Panel

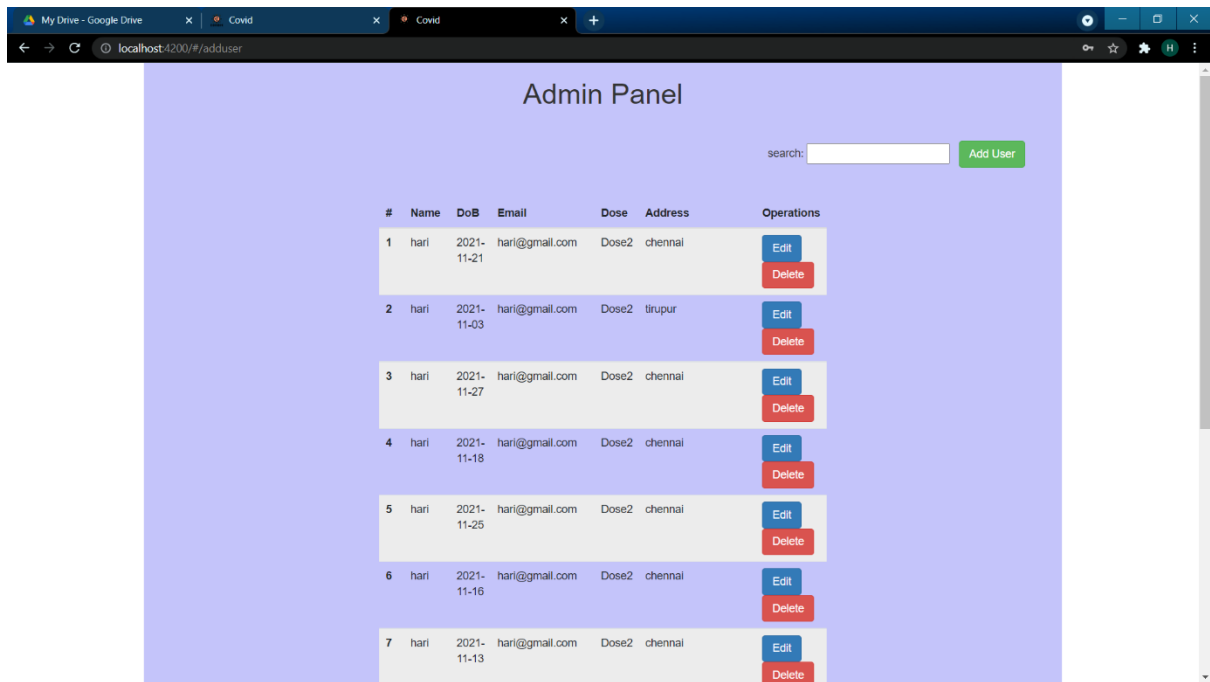
search:

patient detail deleted successfully!!

#	Name	DoB	Email	Dose	Address	Operations
1	hari	2021-11-03	hari@gmail.com	Dose2	tirupur	Edit Delete
2	hari	2021-11-27	hari@gmail.com	Dose2	chennai	Edit Delete
3	hari	2021-11-18	hari@gmail.com	Dose2	chennai	Edit Delete
4	hari	2021-11-25	hari@gmail.com	Dose2	chennai	Edit Delete
5	hari	2021-11-16	hari@gmail.com	Dose2	chennai	Edit Delete
6	hari	2021-11-13	hari@gmail.com	Dose2	chennai	Edit Delete
7	hari	2021-11-05	hari@gmail.com	Dose2	chennai	Edit Delete









Admin

Useremail

Password

Login

My Drive - Google Drive x Covid x Covid x +

localhost:4200/#/adduseradmin

search:

Please select your Role

Admin

Login as user

#	Name	DoB	Email	Dose	Address
1	hari	2021-11-21	hari@gmail.com	Dose2	chennai
2	hari	2021-11-03	hari@gmail.com	Dose2	tirupur
3	hari	2021-11-27	hari@gmail.com	Dose2	chennai
4	hari	2021-11-18	hari@gmail.com	Dose2	chennai
5	hari	2021-11-25	hari@gmail.com	Dose2	chennai
6	hari	2021-11-16	hari@gmail.com	Dose2	chennai
7	hari	2021-11-13	hari@gmail.com	Dose2	chennai
8	hari	2021-11-05	hari@gmail.com	Dose2	chennai
9	harikri	2021-11-20	harikri@gmail.com	Dose2	theni
10	mikki	2009-07-29	harikri@gmail.com	Dose2	guidy,chennai
11	aarush	2015-	hari@gmail.com	Dose1	attur,salem

My Drive - Google Drive x Covid x Covid x +

localhost:4200/#/adduseradmin

search:

Please select your Role

Admin

Login as user

#	Name	DoB	Email	Dose	Address
1	hari	2021-11-21	hari@gmail.com	Dose2	chennai
2	hari	2021-11-03	hari@gmail.com	Dose2	tirupur
3	hari	2021-11-27	hari@gmail.com	Dose2	chennai
4	hari	2021-11-18	hari@gmail.com	Dose2	chennai
5	hari	2021-11-25	hari@gmail.com	Dose2	chennai
6	hari	2021-11-16	hari@gmail.com	Dose2	chennai
7	hari	2021-11-13	hari@gmail.com	Dose2	chennai
8	hari	2021-11-05	hari@gmail.com	Dose2	chennai
9	harikri	2021-11-20	harikri@gmail.com	Dose2	theni
10	mikki	2009-07-29	harikri@gmail.com	Dose2	guidy,chennai
11	aarush	2015-	hari@gmail.com	Dose1	attur,salem

My Drive - Google Drive x Covid x Covid x +

localhost:4200/#/adduseradmin

search:

Please select your Role

#	Name	DoB	Email	Dose	Address
1	hari	2021-11-21	hari@gmail.com	Dose2	chennai
2	hari	2021-11-03	hari@gmail.com	Dose2	tirupur
3	hari	2021-11-27	hari@gmail.com	Dose2	chennai
4	hari	2021-11-18	hari@gmail.com	Dose2	chennai
5	hari	2021-11-25	hari@gmail.com	Dose2	chennai
6	hari	2021-11-16	hari@gmail.com	Dose2	chennai
7	hari	2021-11-13	hari@gmail.com	Dose2	chennai
8	hari	2021-11-05	hari@gmail.com	Dose2	chennai
9	harikri	2021-11-20	harikri@gmail.com	Dose2	theni
10	mikki	2009-07-29	harikri@gmail.com	Dose2	guidy,chennai
11	aarush	2015-	hari@gmail.com	Dose1	attur,salem

My Drive - Google Drive x Covid x Covid x +

localhost:4200/#/district

Go Back

Last updated cases

428

District name

Railway Quarantine

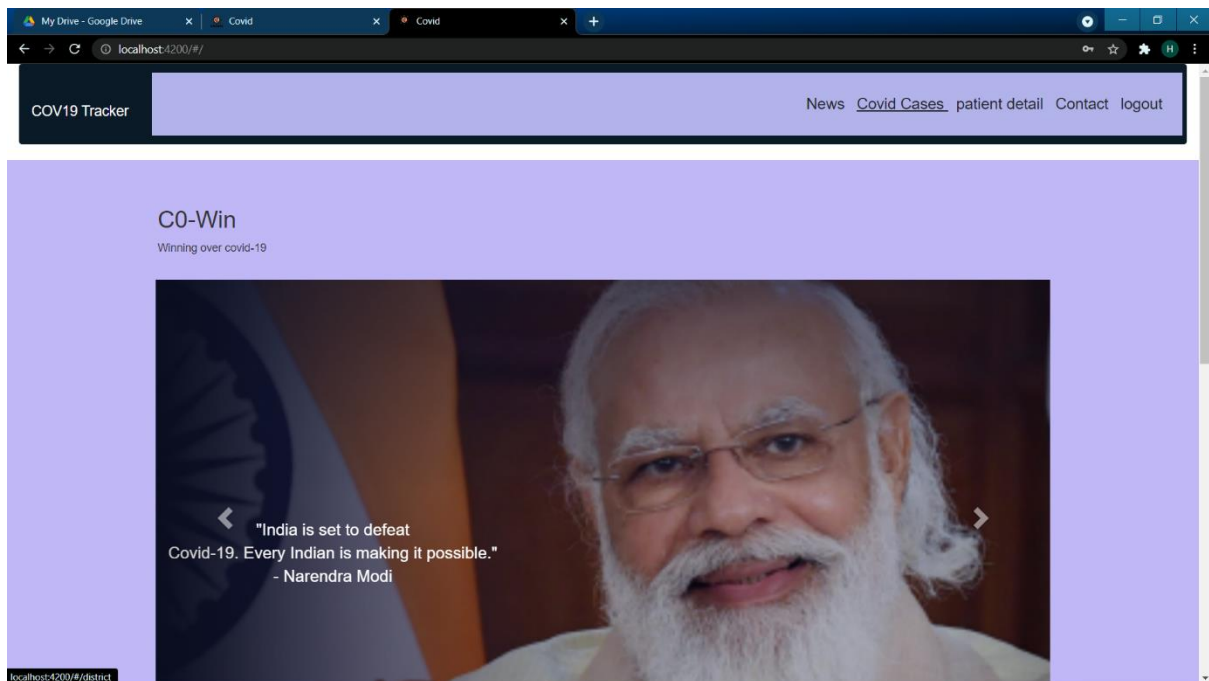
Last updated Recovered cases

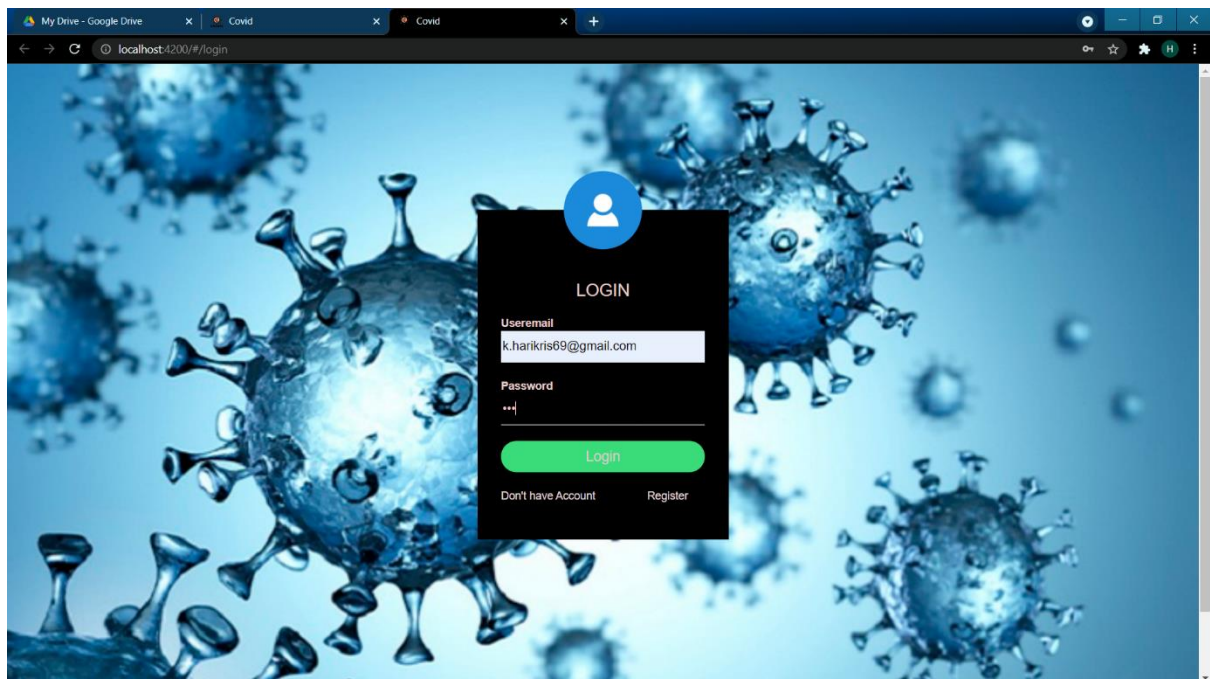
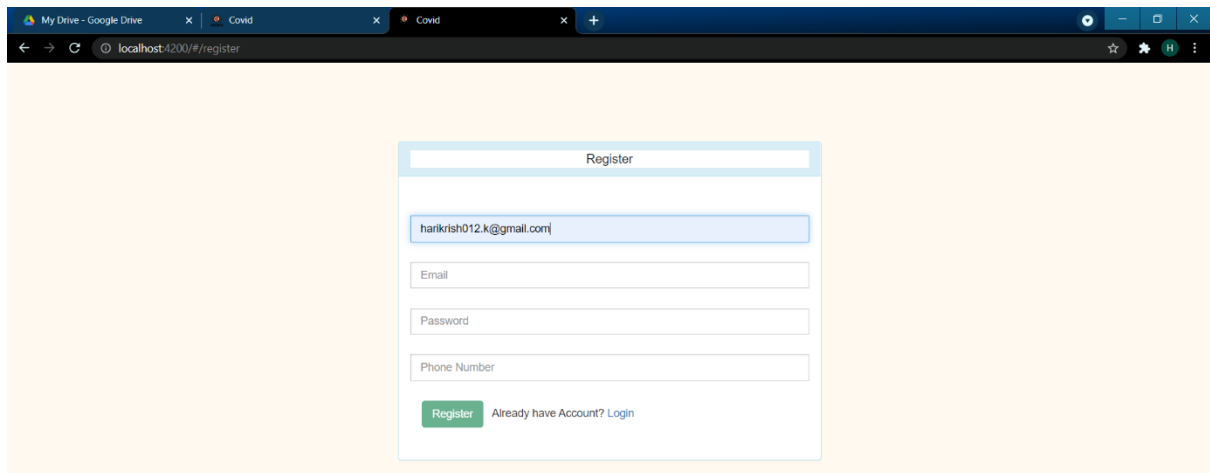
428

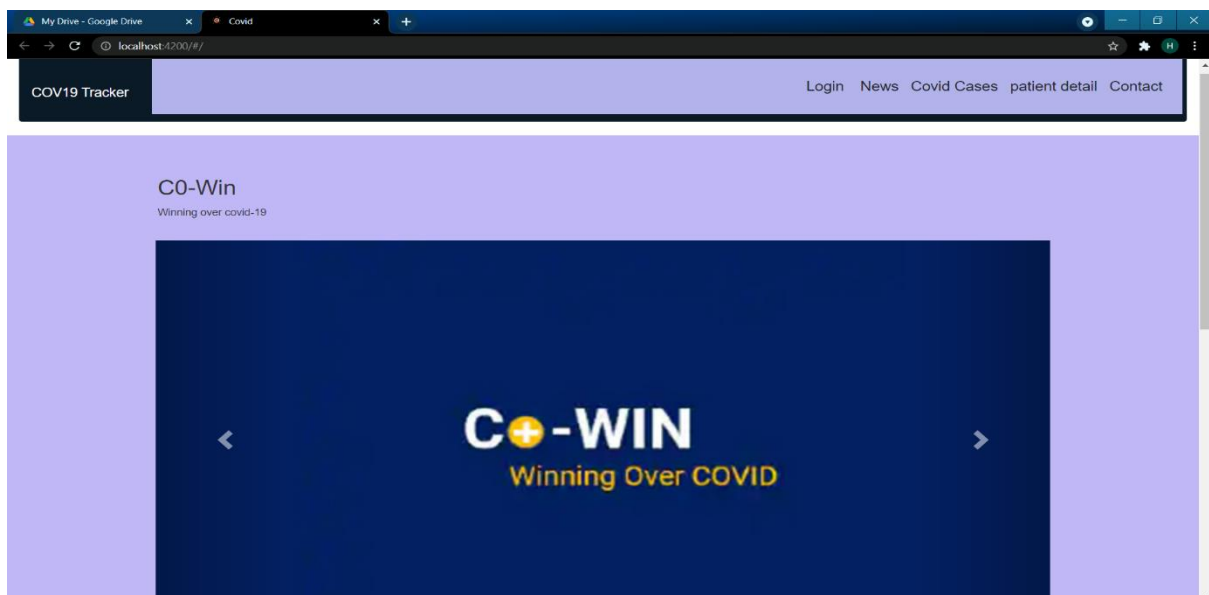
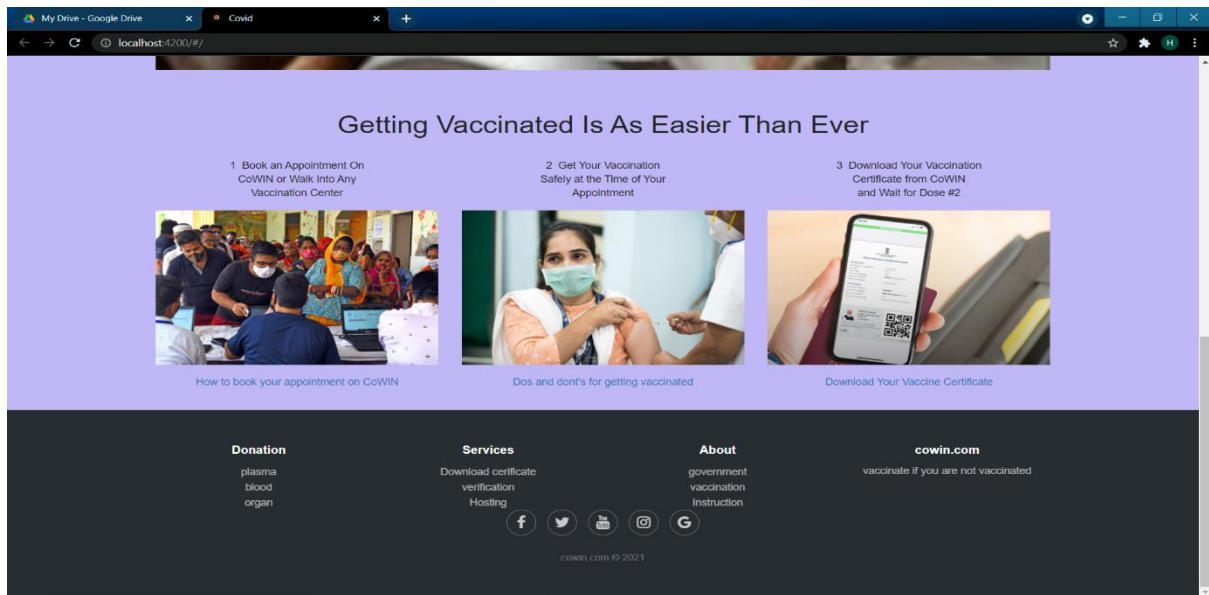
TamilNadu district corona cases

Railway Quarantine

Country	Date	Cases	Recovered	Decreased
Railway Quarantine	17-05-2020	2	0	0
Railway Quarantine	18-05-2020	2	0	0
Railway Quarantine	19-05-2020	2	0	0
Railway Quarantine	20-05-2020	3	0	0
Railway Quarantine	21-05-2020	5	0	0
Railway Quarantine	22-05-2020	5	0	0







```
Windows PowerShell
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\OneDrive\Documents\summa\ss>ng serve -o
Browser application bundle generation complete.

Initial Chunk Files | Names | Size
vendor.js           | vendor | 2.80 MB
polyfills.js        | polyfills | 510.56 kB
styles.css, styles.js | styles | 383.35 kB
main.js             | main | 250.83 kB
runtime.js          | runtime | 6.00 kB
| Initial Total | 3.92 MB

Build at: 2021-11-09T06:38:49.927Z - Hash: 773dee22400sec7f2935 - Time: 17604ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Compiled successfully.
Browser application bundle generation complete.

Initial Chunk Files | Names | Size
main.js             | main | 250.81 kB

4 unchanged chunks

Build at: 2021-11-09T09:55:03.320Z - Hash: 0590227c9200ba8bb2f9 - Time: 2969ms

Compiled successfully.
Browser application bundle generation complete.

5 unchanged chunks

Build at: 2021-11-09T12:02:40.938Z - Hash: 0590227c9200ba8bb2f9 - Time: 2644ms

Compiled successfully.
```

```
C:\Windows\System32\cmd.exe - nodeemon server.js
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\OneDrive\Documents\summa\ss>nodeemon server.js
(nodeemon) 2.0.14
(nodeemon) to restart at any time, enter 'rs'
(nodeemon) watching path(s): *.*
(nodeemon) watching extensions: js,mjs,json
(nodeemon) starting 'node server.js'
check it
(node:21144) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { unifiedTopology: true } to the MongoClient constructor.
(Use 'node --trace-warnings ...' to show where the warning was created)
db created!!
MIDI
{ username: 'k.harikris6@gmail.com', password: '123' }
```