

# Covid19 Data Visualization

## CONTENTS

1. **ABSTRACT**
  2. **INTRODUCTION**
    - 2.1 Overview
    - 2.2 Features
    - 2.3 Why DevOps ?
  3. **SYSTEM CONFIGURATION**
    - 3.1 Operating System
    - 3.2 CPU and RAM
    - 3.3 Language
    - 3.4 Kernel Version
    - 3.5 Database
    - 3.6 DevOps tools
  4. **SOFTWARE DEVELOPMENT LIFE CYCLE**
    - 4.1 Installation
    - 4.2 Tools used in project
    - 4.3 CI/CD Pipeline screenshots
  5. **EXPERIMENTAL SETUP**
    - 5.1 Non Functional Requirements
    - 5.2 Functional Requirements
    - 5.3 Architecture and workflow
    - 5.4 Code walkthrough
    - 5.5 Screenshots and code snippets
  6. **CONCLUSION AND FUTURE WORK**
  7. **REFERENCES**
  8. **LINKS**
-

# 1. ABSTRACT

The recent outbreak of COVID-19 has taken the world by surprise, forcing lockdown and straining public health care systems. COVID-19 is known to be a highly infectious virus, and infected individuals do not initially exhibit symptoms, while some remain asymptomatic. Our main of the project was to make awareness among the people by making Covid19 Data Visualization web app. By visualizing the data using graphical representation of information and data by using visual elements like graph, charts and maps. Data visualization help us to better understand the trend. The web app is build on Angular 11 and data is coming from the API. The data is updated within an interval of 15-30 minutes. The web app show visualization of covid19 disease data

## 2 INTRODUCTION

### 2.1 Overview

Everyone is aware of Covid19 disease One of the most and important times for people and countries who are fighting with this virus. So this is my small initiative to bring awareness among people by showing them statistics of confirmed/active cases of all the states and districts of India through my angular application where data is updated every 15-30 minutes. Our team has made an angular app named **Covid19 Data Visualization**

### 2.2 Features

- Real Time data of the number of cases of COVID-19
- Shows confirmed active cases, recovered cases, and number of deaths
- Shows data from all affected states
- Shows data from the affected districts of states
- Show the India Map which has most affected states
- Show the bar and pie charts for better understanding of data
- Data is updated every 15-30 minutes.

### 2.3 Why Devops ?

DevOps describes a culture and set of processes that bring development and operations teams together to complete software development. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches. And, it's gaining popularity at a rapid rate. DevOps tools consist of configuration management, test and build systems, application deployment, version control and monitoring tools. Continuous integration, continuous delivery and continuous deployment require different tools.

#### 2.3.1. DevOps Features

- Shorter Development Cycles, Faster Innovation
- Reduced Deployment Failures, Rollbacks, and Time to Recover
- Improved Communication and Collaboration
- Increased Efficiencies
- Reduced Costs and IT Headcount

## 3. SYSTEM CONFIGURATION

### 3.1 Operating System

Ubuntu 18.04.04 Bionic Beaver.

## 3.2 CPU and RAM

4 core processor and RAM 8 GB

## 3.3 Language

- Angular 11
- Typescript
- Bootstrap

## 3.4 Kernel Version

Linux Machine 5.4.0-66-generic

## 3.5 Database

we have used the public data API which provide access to all of our data at a national and state level .The data is coming in JSON format links of API used are

- <https://api.covid19india.org/data.json>
- [https://api.covid19india.org/state\\_district\\_wise.json](https://api.covid19india.org/state_district_wise.json)

## 3.6 DevOps Tools

- Source Control Management - GitHub
- Continuous Integration - Jenkins
- Containerization - Docker
- Continuous deployment - Ansible
- Monitoring - ELK Stack (Elastic Search, Logstash, Kibana)

# 4. SOFTWARE DEVELOPMENT LIFE CYCLE

## 4.1 Installation

### Install Node.js

Node.js is an open-source server environment that uses JavaScript on the server. Node.js lets developers use JavaScript to develop wide variety of applications like network applications, command line tools, web API, web applications. We need nodejs for dev tooling (like local web server with live reloading features) and dev experience, we do not need nodejs to run react in production.

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

### Update NPM

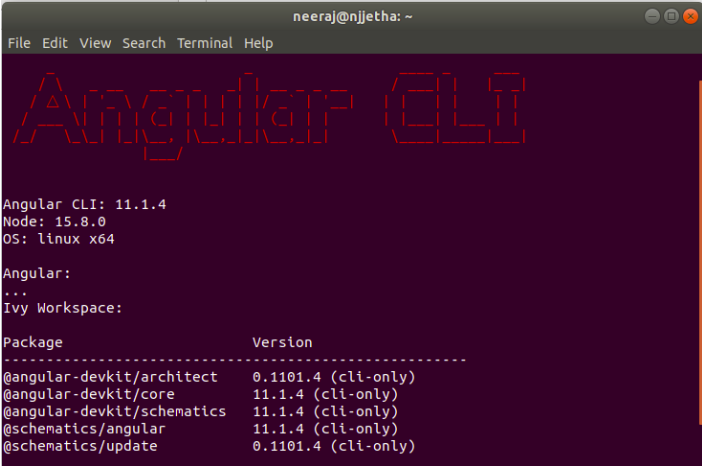
Npm stands for node package manager, it is a dependency management tool for javascript applications. This tool will help to install the libraries and other tools to support angular development

Normally, NPM will be installed with the Node.js itself. However, we can update it to the latest version using the below command.

```
sudo npm install npm@latest -g
```

## Install Angular CLI

```
sudo npm install -g @angular/cli
```

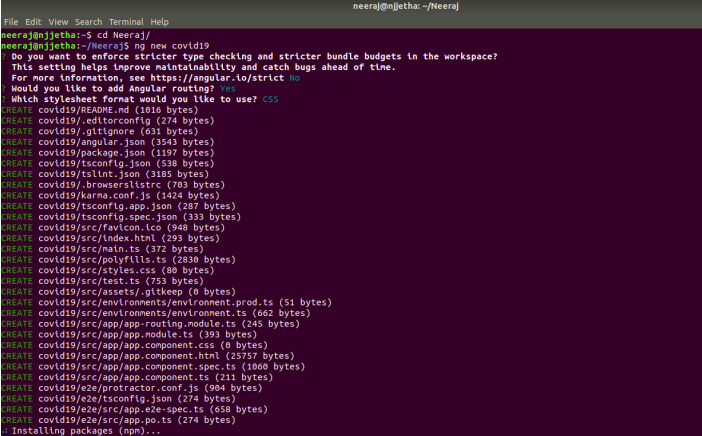


```
neeraj@njetha: ~  
File Edit View Search Terminal Help  
  
Angular CLI  
  
Angular CLI: 11.1.4  
Node: 15.8.0  
OS: linux x64  
  
Angular:  
...  
Ivy Workspace:  
  
Package                                Version  
-----  
@angular-devkit/architect              0.1101.4 (cli-only)  
@angular-devkit/core                   11.1.4 (cli-only)  
@angular-devkit/schematics              11.1.4 (cli-only)  
@schematics/angular                    11.1.4 (cli-only)  
@schematics/update                      0.1101.4 (cli-only)
```

Figure 1 Angular CLI version

## Creating new project

```
ng new covid19
```



```
neeraj@njetha: ~/Neeraj  
File Edit View Search Terminal Help  
neeraj@njetha:~$ cd Neeraj/  
neeraj@njetha:~/Neeraj$ ng new covid19  
Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?  
This setting helps improve maintainability and catch bugs ahead of time.  
For more information, see https://angular.io/strict No  
Would you like to add Angular routing? Yes  
Which stylesheet format would you like to use? CSS  
CREATE covid19/README.md (1016 bytes)  
CREATE covid19/.editorconfig (274 bytes)  
CREATE covid19/.gitignore (631 bytes)  
CREATE covid19/angular.json (3543 bytes)  
CREATE covid19/package.json (1197 bytes)  
CREATE covid19/tsconfig.json (538 bytes)  
CREATE covid19/tslint.json (3185 bytes)  
CREATE covid19/.browserslistrc (103 bytes)  
CREATE covid19/karma.conf.js (1424 bytes)  
CREATE covid19/tsconfig.app.json (287 bytes)  
CREATE covid19/tsconfig.spec.json (333 bytes)  
CREATE covid19/src/favicon.ico (948 bytes)  
CREATE covid19/src/index.html (293 bytes)  
CREATE covid19/src/main.ts (372 bytes)  
CREATE covid19/src/polyfills.ts (2830 bytes)  
CREATE covid19/src/styles.css (88 bytes)  
CREATE covid19/src/test.ts (751 bytes)  
CREATE covid19/src/assets/.gitkeep (0 bytes)  
CREATE covid19/src/environments/environment.prod.ts (51 bytes)  
CREATE covid19/src/environments/environment.ts (662 bytes)  
CREATE covid19/src/app/app-routing.module.ts (245 bytes)  
CREATE covid19/src/app/app.module.ts (393 bytes)  
CREATE covid19/src/app/app.component.css (0 bytes)  
CREATE covid19/src/app/app.component.html (25757 bytes)  
CREATE covid19/src/app/app.component.spec.ts (1868 bytes)  
CREATE covid19/src/app/app.component.ts (211 bytes)  
CREATE covid19/e2e/protractor.conf.js (984 bytes)  
CREATE covid19/e2e/tsconfig.json (274 bytes)  
CREATE covid19/e2e/src/app.e2e-spec.ts (658 bytes)  
CREATE covid19/e2e/src/app.po.ts (274 bytes)  
Installing packages (npm)...
```

Figure 2 Creating new project

## Commands used to generate components

```
ng g c <name>
```

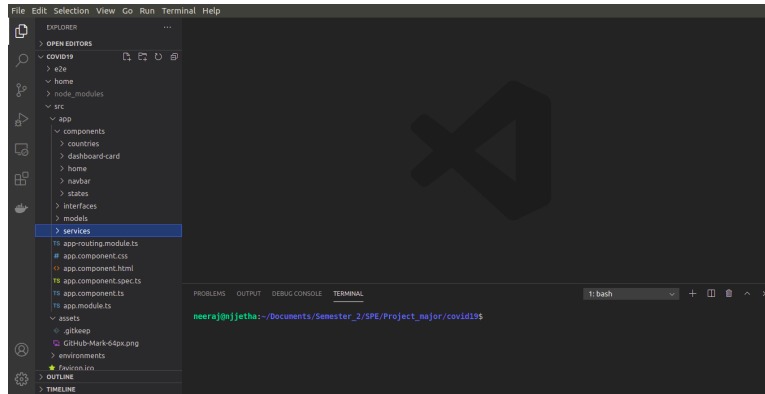


Figure 3 creating all the components

### Commands to generate interface

```
ng g interface <name> <type>
```

### Commands to generate service

```
ng g service <name>
```

## 4.2 Tools used in Project

**Github:-** A Source Code Management (SCM) is a software tool used by programmers to manage the source codes. For our project every team member would clone the repository from github. Create a different branch locally on their system and then merge it with master after pulling the latest code from git, resolving any conflicts and then push the changes to git.

- **git clone <repository url>-** This command copies the entire data on the git url
- **git checkout -b <branch\_name>-** This command creates a new branch with the name as in 'branch\_name'
- **git add <changed files>-** This command adds changes in the working directory to the staging area
- **git commit -m "message while committing"-** This command is used to save your changes to the local repository with -m used to provide a concise description that helps your teammates (and yourself) understand what happened.
- **git checkout master-** This command switches to master branch
- **git pull-** This command is used to update the local version of a repository from a remote.
- **git merge <branch\_name> -** This command is used to integrate changes from another branch.
- **git push-** This command will push all the latest code to the repository

```

commit 2c0d6592247d1f8f69fc177fb45cefa6fe57b344 (HEAD -> master, origin/master)
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Wed Apr 28 00:38:42 2021 +0530

    Frontend Improved

commit cbfc9ec03f4f23774bbdc6886a78d962fa9c1e2e
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Tue Apr 27 16:07:34 2021 +0530

    move the dockerfile

commit 5c1ef786743399176f8631a46a0bf54eb3aba9aa
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Tue Apr 27 15:47:10 2021 +0530

    added docker file

commit 6d4fe80a151e3d224596cac65220530f7a28017a
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Tue Apr 27 00:40:12 2021 +0530

    improvement in cards

commit bdc2fbec609f034cbd6f730981821f79816b353e
Merge: 9d01e47 e183a1c
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Sat Apr 17 08:29:37 2021 +0530

    Merge branch 'master' of https://github.com/njjetha/Covid19-Data-Visualization

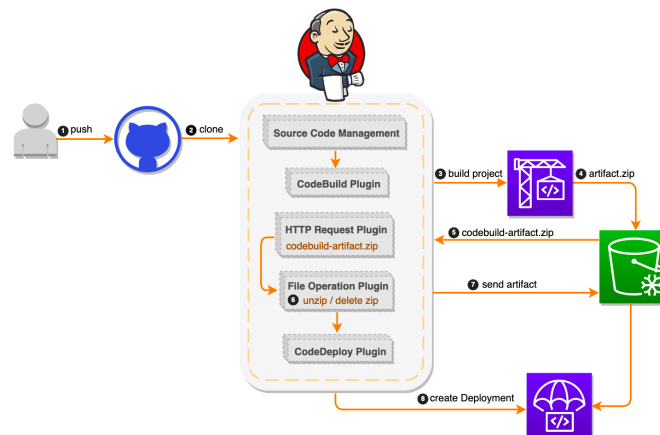
    Geochart added

commit 9d01e4777863cc26d7cfebb95c45dc8d0b8ec309
Author: njjetha <neeraj.jetha@iitb.org>
Date:   Sat Apr 17 08:29:29 2021 +0530

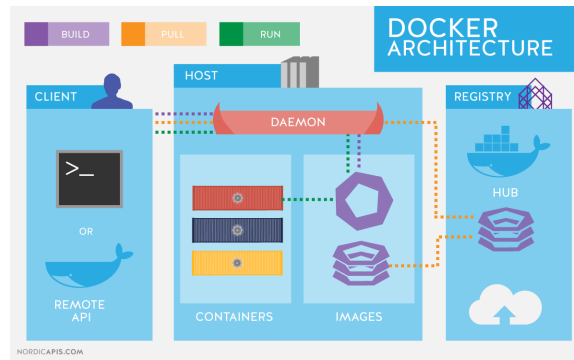
```

Figure 4 git logs

**Jenkins** :- Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment technologies.

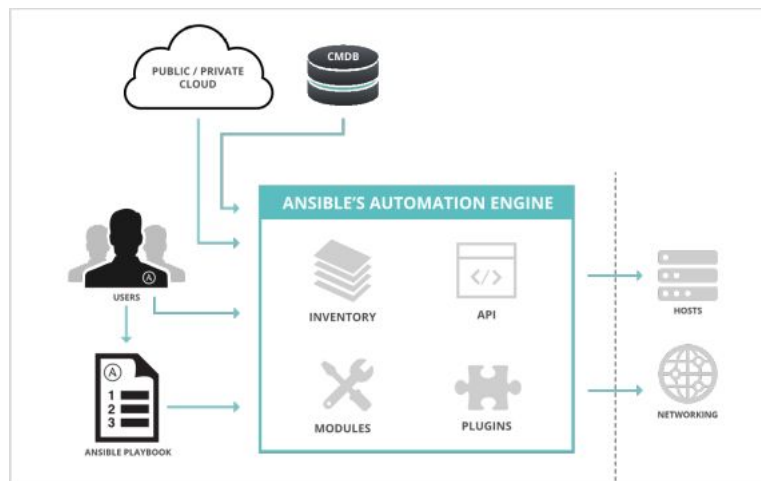


**Docker**: Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package



**Ansible :-** Open Source automation platform. Automation Engine that runs ansible playbooks, playbooks are defined tasks, where we define environments and workflows. There are two types of machines in the Ansible architecture: control nodes and managed hosts. Ansible is installed and run from a control node, and this machine also has copies of your Ansible project files.

Each task runs a module, a small piece of code (written in Python, PowerShell, or some other language), with specific arguments. Each module is essentially a tool in your toolkit. Ansible ships with hundreds of useful modules that can perform a wide variety of automation tasks. They can act on system files, install software, or make API calls.



**ELK :-** "ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.



## 4.3 CI/CD Pipeline Screenshots

### Pipeline Script

```

pipeline {
  agent any
  environment{
    imageName=""
  }
  stages {
    stage('git cloning') {
      steps {
        script{
          git 'https://github.com/njjetha/Covid19-Data-Visualization'
        }
      }
    }

    stage('Docker Build to image') {
      steps {
        script{
          imageName=docker.build "njjetha/covid19:latest"
        }
      }
    }

    stage('Push Docker image') {
      steps {
        script{
          docker.withRegistry('', 'docker-jenkins'){
            imageName.push()
          }
        }
      }
    }

    stage('Ansible Pull Docker image') {
      steps {
        ansiblePlaybook becomeUser: null, colored: true, disableHostKeyChecking: true,
          installation:'Ansible',inventory: 'remote-server', playbook: 'covid19-playbook.yml', sudoUser: null
      }
    }
  }
}

```

#### **Stage: 1 → Git Cloning**

In this stage the code is cloned from the github repo

#### **Stage: 2 → Docker Build to image**

In this stage we create the docker image where we first install all the dependencies required to run the docker image

#### **Step 3: → Push Docker image**

In this stage we push the docker image to docker hub by providing the credential of docker hub

#### **Step 4: → Pulling Docker image**

In this stage we pull the docker image on the local host using ansible.



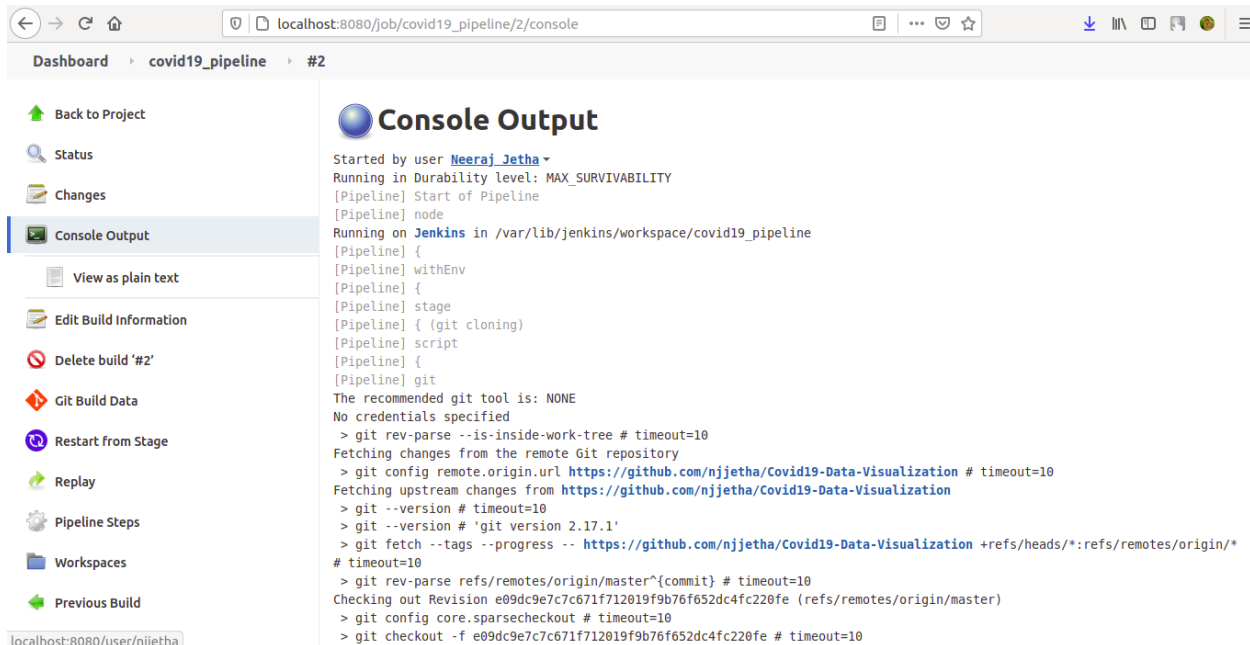


Figure 5 Git Cloning

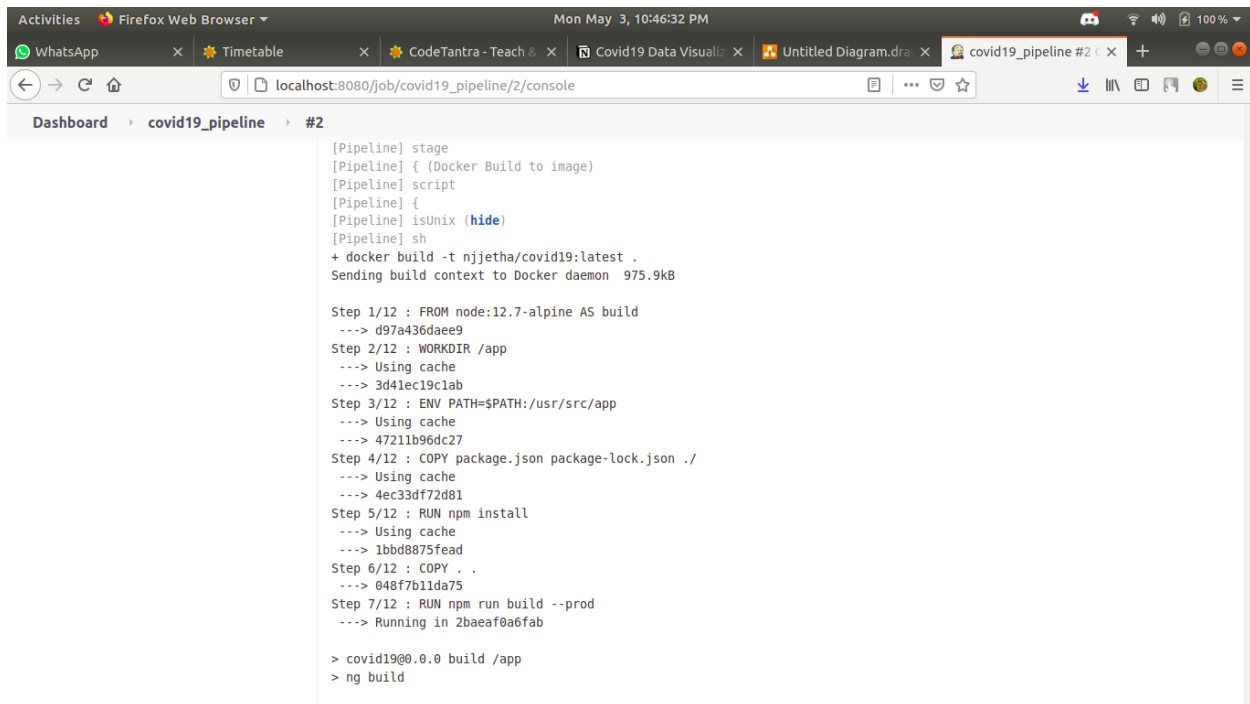
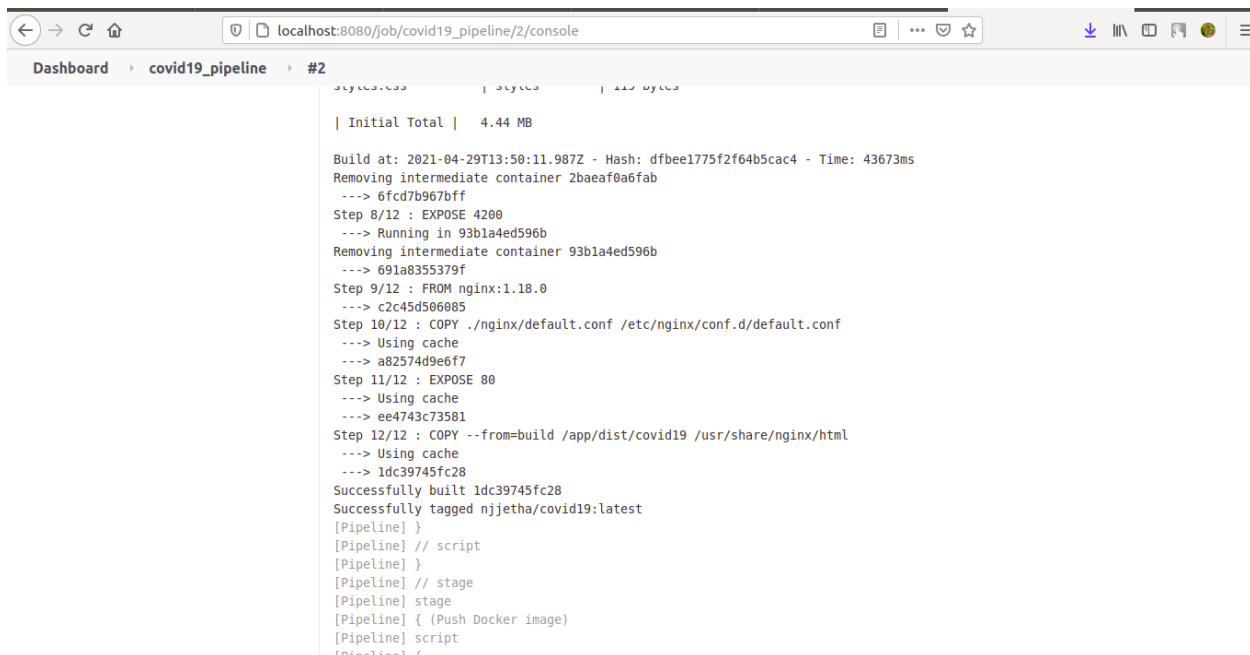
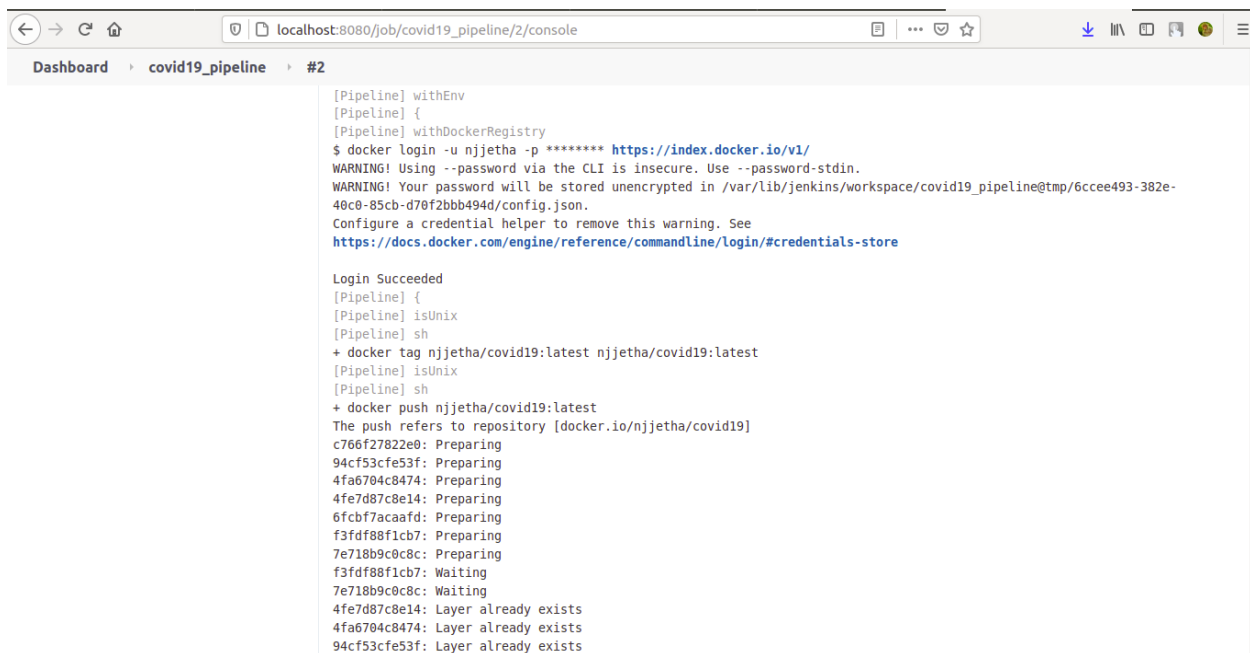


Figure 6 Building Docker image



```
Dashboard > covid19_pipeline > #2  
[Pipeline] }  
[Pipeline] // script  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Push Docker image)  
[Pipeline] script  
[Pipeline] {  
  | Initial Total | 4.44 MB  
  
  Build at: 2021-04-29T13:50:11.987Z - Hash: dfbeel775f2f64b5cac4 - Time: 43673ms  
  Removing intermediate container 2baeaf0a6fab  
  ---> 6fcd7b967bff  
  Step 8/12 : EXPOSE 4200  
  ---> Running in 93b1a4ed596b  
  Removing intermediate container 93b1a4ed596b  
  ---> 691a8355379f  
  Step 9/12 : FROM nginx:1.18.0  
  ---> c2c45d506085  
  Step 10/12 : COPY ./nginx/default.conf /etc/nginx/conf.d/default.conf  
  ---> Using cache  
  ---> a82574d9e6f7  
  Step 11/12 : EXPOSE 80  
  ---> Using cache  
  ---> ee4743c73581  
  Step 12/12 : COPY --from=build /app/dist/covid19 /usr/share/nginx/html  
  ---> Using cache  
  ---> 1dc39745fc28  
  Successfully built 1dc39745fc28  
  Successfully tagged njjetha/covid19:latest  
[Pipeline] }  
[Pipeline] }  
[Pipeline] }  
[Pipeline] }  
[Pipeline] }
```

Figure 7 Pushing Docker image



```
Dashboard > covid19_pipeline > #2  
[Pipeline] withEnv  
[Pipeline] {  
[Pipeline] withDockerRegistry  
$ docker login -u njjetha -p ***** https://index.docker.io/v1/  
WARNING! Using --password via the CLI is insecure. Use --password-stdin.  
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/covid19_pipeline/tmp/6ccee493-382e-40c0-85cb-d70f2bbb494d/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store  
  
Login Succeeded  
[Pipeline] {  
[Pipeline] isUnix  
[Pipeline] sh  
+ docker tag njjetha/covid19:latest njjetha/covid19:latest  
[Pipeline] isUnix  
[Pipeline] sh  
+ docker push njjetha/covid19:latest  
The push refers to repository [docker.io/njjetha/covid19]  
c766f27822e0: Preparing  
94cf53cfe53f: Preparing  
4fa6704c8474: Preparing  
4fe7d87c8e14: Preparing  
6fcbf7acaaaf: Preparing  
f3fdf88f1cb7: Preparing  
7e718b9c0c8c: Preparing  
f3fdf88f1cb7: Waiting  
7e718b9c0c8c: Waiting  
4fe7d87c8e14: Layer already exists  
4fa6704c8474: Layer already exists  
94cf53cfe53f: Layer already exists  
[Pipeline] }
```

Figure 8 Docker Hub login successful and pushing image

```

[Pipeline] ansiblePlaybook
[covid19_pipeline] $ /usr/bin/ansible-playbook covid19-playbook.yml -i remote-server

PLAY [Pull docker image of covid19] *****

TASK [Gathering Facts] *****
[0;35m[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.1.1 should use [0m
[0;35m/usr/bin/python3, but is using /usr/bin/python for backward compatibility with [0m
[0;35mprior Ansible releases. A future Ansible release will default to using the [0m
[0;35mdiscovered platform python for this host. See https://docs.ansible.com/ansible/[0m
[0;35m2.9/reference_appendices/interpreter_discovery.html for more information. This [0m
[0;35mfeature will be removed in version 2.12. Deprecation warnings can be disabled [0m
[0;35mby setting deprecation_warnings=False in ansible.cfg.[0m
[0;32mok: [127.0.1.1][0m

TASK [Pull covid19 devops image] *****
[0;32mok: [127.0.1.1][0m

PLAY RECAP *****
[0;32m127.0.1.1: [0;32mok=2 [0m changed=0 unreachable=0 failed=0 skipped=0 rescued=0
ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Figure 9 Docker image pulling through Ansible

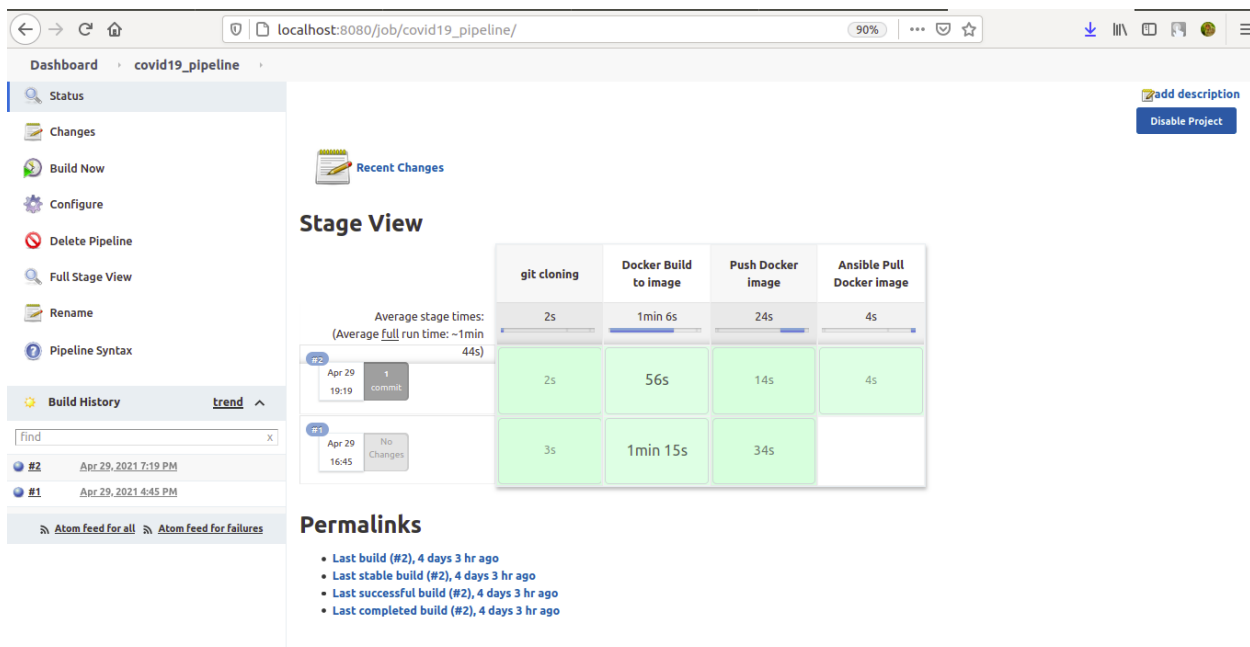


Figure 10 Stages of Pipeline

- **jasmine-core.** Jasmine is the framework we are going to use to create our tests. It has a bunch of functionalities to allow us the write different kinds of tests.
- **karma.** Karma is a task runner for our tests. It uses a configuration file in order to set the startup file, the reporters, the testing framework, the browser among other things.

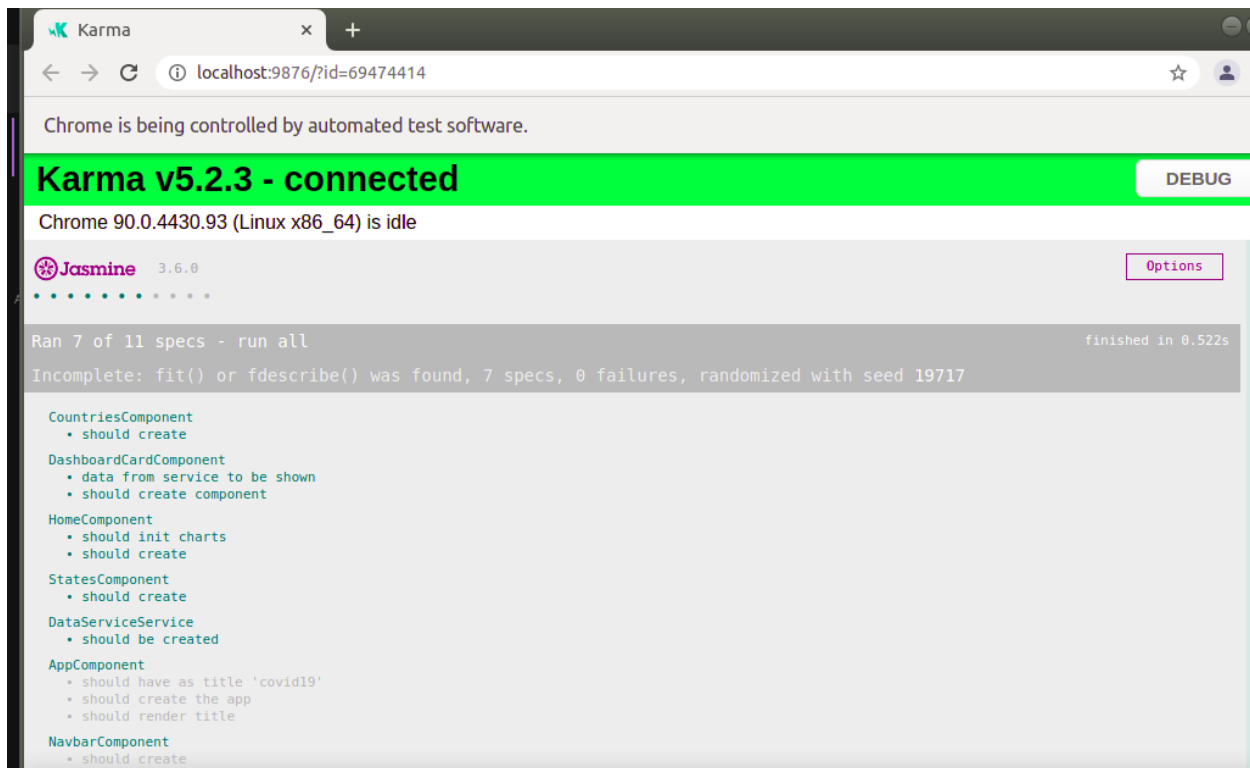


Figure 11 Testing

```
| Initial Total | 4.78 MB
Build at: 2021-05-03T17:27:42.688Z - Hash: 2969ad6cd22acd5ffc37 - Time: 19189ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
^C
neeraj@njetha:~/Documents/Semester_2/SPE/Project_major/covid19$ ng test
Compiling @angular/core/testing : es2015 as esm2015
Compiling @angular/compiler/testing : es2015 as esm2015
Compiling @angular/platform-browser/testing : es2015 as esm2015
Compiling @angular/common/testing : es2015 as esm2015
Compiling @angular/common/http/testing : es2015 as esm2015
Compiling ngx-logger : es2015 as esm2015
Compiling @angular/platform-browser-dynamic/testing : es2015 as esm2015
Compiling @angular/router/testing : es2015 as esm2015
" Generating browser application bundles...03 05 2021 23:13:10.767:WARN [karma]: No captured browser, open http://localhost:9876/
03 05 2021 23:13:10.776:INFO [karma-server]: Karma v5.2.3 server started at http://localhost:9876/
03 05 2021 23:13:10.776:INFO [launcher]: Launching browsers Chrome with concurrency unlimited
" Generating browser application bundles (phase: building)...03 05 2021 23:13:10.784:INFO [launcher]: Starting browser Chrome
✓ Browser application bundle generation complete.
" Generating browser application bundles...03 05 2021 23:13:20.856:WARN [karma]: No captured browser, open http://localhost:9876/
✓ Browser application bundle generation complete.
03 05 2021 23:13:21.317:INFO [Chrome 90.0.4430.93 (Linux x86_64)]: Connected on socket p9z2NZ0ZywmnMlyQAAAA with id 69474414
LOG: undefined
Chrome 90.0.4430.93 (Linux x86_64): Executed 1 of 11 SUCCESS (0 secs / 0.057 secs)
Chrome 90.0.4430.93 (Linux x86_64): Executed 7 of 11 (skipped 4) SUCCESS (0.527 secs / 0.388 secs)
```

Figure 12 Testing file running

**NGX Logger** is a simple logging module for angular (currently supports angular 6+). It offers the following features:

- Pretty printing to the console
- Enable logging based on the level specified

- Control log level based on the current environment
- Send log messages to server via HTTP for centralised logging
- Indicates log location and line number

2021-05-10T16:41:51.670Z DEBUG [main.js:553]	ngx-logger.js:753
Navbar Component working correctly	
Angular is running in development mode. Call enableProdMode() to enable production mode.	
2021-05-10T16:41:51.699Z DEBUG [main.js:934]	ngx-logger.js:753
State Component working correctly	
2021-05-10T16:41:51.708Z DEBUG [main.js:452]	ngx-logger.js:753
Dashboard Component working correctly	
[Violation] 'load' handler took 195ms	zone-evergreen.js:1611
2021-05-10T16:41:52.179Z DEBUG [main.js:1003]	ngx-logger.js:753
DailyConfirmed data came ->366455	
[WDS] Live Reloading enabled.	client:52
[Violation] 'load' handler took 488ms	zone-evergreen.js:1611
▶ [Violation] Added non-passive event listener to a scroll-blocking 'mousewheel' event. Consider marking event handler as 'passive' to make the page more responsive. See <a href="https://www.chromestatus.com/feature/5745543795965952">https://www.chromestatus.com/feature/5745543795965952</a>	

Figure 12 Logs at browser

top	Filter	All levels	No Issues
2021-05-13T03:14:00.615Z DEBUG [main.js:568]	Navbar Component working correctly	vendor.js:sourcemap:16934	
Angular is running in development mode. Call enableProdMode() to enable production mode.			
2021-05-13T03:14:00.643Z DEBUG [main.js:957]	State Component working correctly	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.647Z DEBUG [main.js:467]	Dashboard Component working correctly	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.672Z DEBUG [main.js:696]	Arranging the data in tabular format and specifying style	vendor.js:sourcemap:16934	
✖ 2021-05-13T03:14:00.674Z ERROR [main.js:697]	Table format	vendor.js:sourcemap:16923	
2021-05-13T03:14:00.679Z DEBUG [main.js:721]	Arranging the data in sub tabular format and specifying style	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.680Z DEBUG [main.js:742]	Fetching Covid19 Data from the api	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.685Z DEBUG [main.js:746]	Fetching Covid19 Data statewide from the api	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.689Z DEBUG [main.js:742]	Fetching Covid19 Data from the api	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.693Z DEBUG [main.js:1037]	Line Chart function working	vendor.js:sourcemap:16934	
2021-05-13T03:14:00.695Z DEBUG [main.js:1071]	Linegraph data length ->1	vendor.js:sourcemap:16934	
✖ 2021-05-13T03:14:00.765Z ERROR [vendor.js:13792]	FAILED TO LOG ON SERVER: Table format	vendor.js:sourcemap:16923	
▶ HttpErrorResponse {headers: HttpHeaders, status: 404, statusText: "Not Found", url: "http://localhost:4200/api/logs", ok: false, ...}			
[WDS] Live Reloading enabled.		vendor.js:sourcemap:21607	
2021-05-13T03:14:00.922Z INFO [main.js:969]	covid data recieved from api is not null	vendor.js:sourcemap:16926	
[Violation] 'load' handler took 290ms		polyfills.js:sourcemap:1706	
2021-05-13T03:14:01.261Z INFO [main.js:986]	State wise covid data recieved is not null	vendor.js:sourcemap:16926	
2021-05-13T03:14:01.262Z INFO [main.js:1007]	StateName is State_Unassigned. District name is Unassigned. Confirmed cases 0	vendor.js:sourcemap:16926	

Figure 13 Logs at different level are of different color

[WDS] Live Reloading enabled.	vendor.js:21607
2021-05-13T03:14:00.922Z INFO [main.js:969] covid data recived from api is not null	vendor.js:16926
[Violation] 'load' handler took 290ms	polyfills.js:1706
2021-05-13T03:14:01.261Z INFO [main.js:986] State wise covid data recieved is not null	vendor.js:16926
2021-05-13T03:14:01.262Z INFO [main.js:1007] StateName is State Unassigned District name is Unassigned Confirmed cases 0	vendor.js:16926
2021-05-13T03:14:01.263Z INFO [main.js:1007] StateName is Andaman and Nicobar Islands District name is Nicobars Confirmed cases 0	vendor.js:16926
2021-05-13T03:14:01.264Z INFO [main.js:1007] StateName is Andaman and Nicobar Islands District name is North and Middle Andaman Confirmed cases 1	vendor.js:16926
2021-05-13T03:14:01.264Z INFO [main.js:1007] StateName is Andaman and Nicobar Islands District name is South Andaman Confirmed cases 51	vendor.js:16926
2021-05-13T03:14:01.265Z INFO [main.js:1007] StateName is Andaman and Nicobar Islands District name is Unknown Confirmed cases 6418	vendor.js:16926
2021-05-13T03:14:01.266Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Foreign Evacuees Confirmed cases 434	vendor.js:16926
2021-05-13T03:14:01.267Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Anantapur Confirmed cases 103593	vendor.js:16926
2021-05-13T03:14:01.267Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Chittoor Confirmed cases 145814	vendor.js:16926
2021-05-13T03:14:01.268Z INFO [main.js:1007] StateName is Andhra Pradesh District name is East Godavari Confirmed cases 169999	vendor.js:16926
2021-05-13T03:14:01.269Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Guntur Confirmed cases 125852	vendor.js:16926
2021-05-13T03:14:01.270Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Krishna Confirmed cases 72569	vendor.js:16926
2021-05-13T03:14:01.271Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Kurnool Confirmed cases 95877	vendor.js:16926
2021-05-13T03:14:01.271Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Other State Confirmed cases 2461	vendor.js:16926
2021-05-13T03:14:01.272Z INFO [main.js:1007] StateName is Andhra Pradesh District name is Prakasam Confirmed cases 85929	vendor.js:16926

Figure 14 Logs at different level are of different color

**njjetha / covid19**

*This repository does not have a description*

🕒 Last pushed: 4 days ago

**Docker commands**

To push a new tag to this repository,

```
docker push njjetha/covid19:tagname
```

[Public View](#)

**Tags and Scans**

🔒 VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		3 days ago	4 days ago

[See all](#)

**Recent builds**

[Link a source provider and run a build to see build results here.](#)

Figure 15 Docker Hub image

A **Dockerfile** is a text file that Docker reads in from top to bottom. It contains a bunch of instructions which informs Docker HOW the Docker image should get built.

```

1  #
2
3  ### STAGE 1: Build ###
4  FROM node:12.7-alpine AS build
5  WORKDIR /app
6  ENV PATH=$PATH:/usr/src/app
7  COPY package.json package-lock.json ./
8  RUN npm install
9  COPY . .
10 RUN npm run build --prod
11 EXPOSE 4200
12
13
14
15
16 ### STAGE 2: Run ###
17 FROM nginx:1.18.0
18 COPY ./nginx/default.conf /etc/nginx/conf.d/default.conf
19 EXPOSE 80
20 COPY --from=build /app/dist/covid19 /usr/share/nginx/html
21

```

Figure 16 Dockerfile

**Playbooks** are one of the core features of Ansible and tell Ansible what to execute. Playbooks contain the steps which the user wants to execute on a particular machine. Playbooks are run sequentially. Playbooks are the building blocks for all the use cases of Ansible.

```

! covid19-playbook.yml
1  ---
2  - name: Pull docker image of covid19
3    hosts: all
4    tasks:
5      - name: Pull covid19 devops image
6        docker_image:
7          name: njjetha/covid19
8          source: pull

```

Figure 17 Ansible playbook

Command to run docker image

```
docker run -p 4200:80 --name covid19 njjetha/covid19
```

```
Activities Terminal Mon May 3, 10:46:00 PM neeraj@njetha: ~
neeraj@njetha:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
njetha/covid19       latest             1dc39745fc28       4 days ago         142MB
<none>               <none>            59a9b460acd6       4 days ago         712MB
<none>               <none>            078ac91ef21a       6 days ago         712MB
nginx                1.18.0             c2c45d506085       3 weeks ago        133MB
njetha/calculator-devops latest             9c3f0c38e100       7 weeks ago        516MB
njetha/calculator-devops <none>            0a88aefa05af       7 weeks ago        516MB
ubuntu-sleeper       latest             1ce51f10b74b       2 months ago       72.9MB
njetha/ubuntu-sleeper latest             1ce51f10b74b       2 months ago       72.9MB
tomcat               latest             bf4709e77b18       2 months ago       667MB
openjdk              8                  9324460525ca       2 months ago       514MB
maven                latest             3956fab279d0       3 months ago       753MB
ubuntu               latest             f63181f19b2f       3 months ago       72.9MB
hello-world          latest             bf756fb1ae65       16 months ago      13.3kB
node                 12.7-alpine       d97a436dae9        21 months ago      79.3MB
jenkins              2.60.3            cd14cecfdb3a       2 years ago        696MB
neeraj@njetha:~$
```

Figure 18 Docker images

## 5. EXPERIMENTAL SETUP

### 5.1 Non Functional Requirements

#### Usability

Being an app centered around visualization and the target audience being the everyday user of the internet, the interface should be graphical and very intuitive to use. The users should be able to figure out the different features and available customization with little use.

#### Security

The app should rely on anonymous data that is available to the public. It should not reveal the private information of any individuals.

#### Performance

The app should be responsive. The visualizations to be displayed should be loaded and rendered quickly and should change dynamically based on the users actions.

#### Scalability

The app should be able to scale to deal with large amount of data without impacting the performance. If the application is modified from client side to server side then the server should be able to handle the users traffic without a drop in performance.

### 5.2 Functional Requirements

- The application should display time series COVID-19 data using plots.
- It should use a live data source that is updated within an interval of 15-30 minutes.
- The application should show the confirmed, recovered, and death cases separately.
- The graph and map automatically updated while clicking on the bullet of cases



- The state table should be expanded on click on the row for showing district cases .

## 5.3 Architecture and Workflow

Here in the reference design diagram of our application and how the data flow .Data coming from the public API in the json format

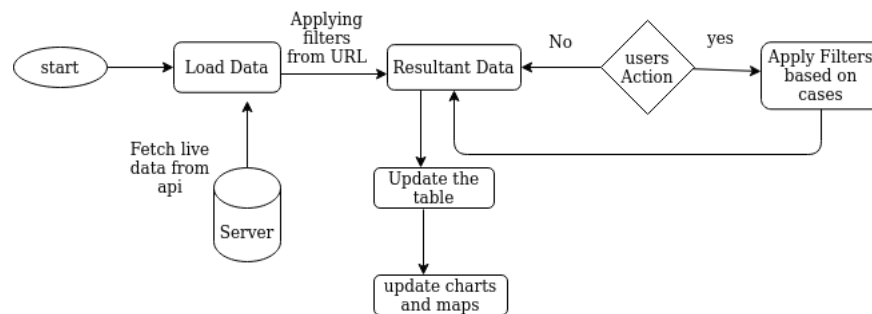


Figure 19 Workflow

## 5.4 Code Walkthrough

There are 4 components , one service file and one interface

### Components

- **AppComponent:** This is first component which runs and calls the `NavbarComponent`
- **NavbarComponent :-** This is component which contains links on the navbar
- **HomeComponent :** This component is used to display the charts and maps of different cases
- **DashboardComponent :-** This component contains cards of different cases which are updated within an interval of 15-30 minutes
- **stateComponent:** Contains a list of the districts for specific states and loaded dynamically. On click of a particular state.
- **dataservice :** It is used for fetching data from the API
- **dataInterface :** It is used to define the structure of the data coming from the API.

### API for fetching data

- **Daily-Indian-Case:** <https://api.covid19india.org/data.json>
- **State-district-wise :** [https://api.covid19india.org/state\\_district\\_wise.json](https://api.covid19india.org/state_district_wise.json)

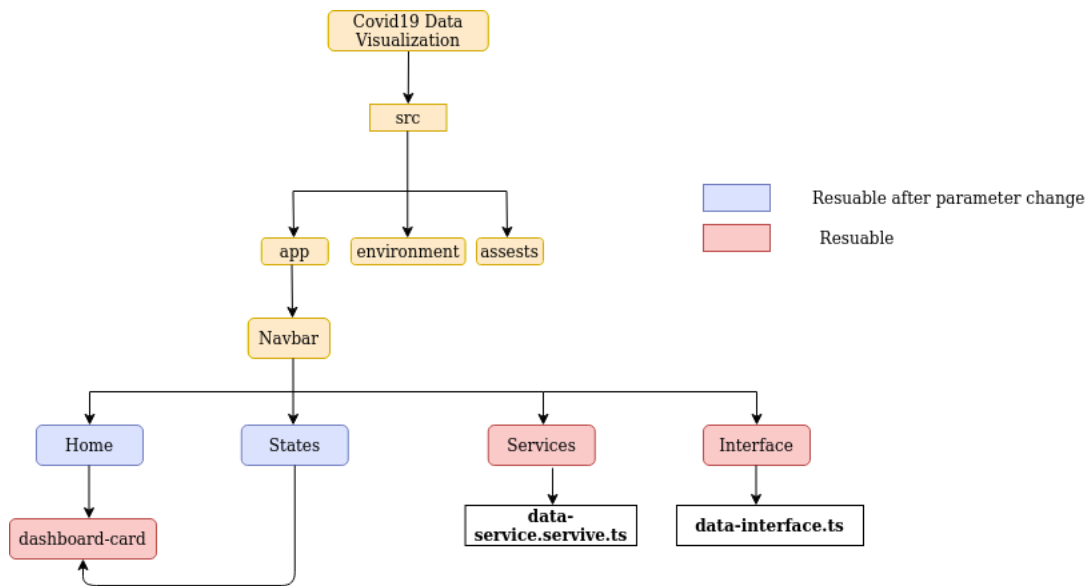


Figure 20 CodeFlow

## 5.5 Screenshots and code snippets

```

1  import { HttpClient } from '@angular/common/http';
2  import { Injectable } from '@angular/core';
3  import { map } from 'rxjs/operators';
4  import { CovidData } from '../interfaces/data-interface';
5  import { CovidDataSummary } from '../models/covidData';
6
7  @Injectable({
8    providedIn: 'root'
9  })
10 export class DataServiceService {
11
12   private globalDataUrl = 'https://api.covid19india.org/state_district_wise.json';
13   constructor(private http: HttpClient) {}
14
15   stateName: any[] = []; //It will hold the stateName
16
17   getCovidData() {
18     return this.http.get(this.globalDataUrl, { responseType: 'json' }).pipe(
19       map(result => {
20         //used to display the state value as key
21
22         for (var key in result) {
23           if (result.hasOwnProperty(key)) {
24             this.stateName.push(key);
25             // console.log(key);
26           }
27         }
28         this.stateName.splice(0, 1);
29         // console.log(this.stateName[0]);
30       })
31     );
32   }
33 }
  
```

Figure 21 dataservice.service.ts

```

1  public getTreeTableColumns() {
2      const columns = [
3          {
4              field: 'district', header: 'DISTRICT', style: { width: '100px', 'min-width': '75px', 'max-width': '100px' },
5              type: 'text'
6          },
7          {
8              field: 'confirmed', header: 'CONFIRMED', style: { width: '100px', 'min-width': '75px', 'max-width': '100px' },
9              type: 'text'
10         },
11         {
12             field: 'active', header: 'ACTIVE', style: { width: '100px', 'min-width': '75px', 'max-width': '100px' },
13             type: 'text'
14         },
15         {
16             field: 'recovered', header: 'RECOVERED', style: { width: '100px', 'min-width': '75px', 'max-width': '100px' },
17             type: 'text'
18         },
19         {
20             field: 'deceased', header: 'DECEASED', style: { width: '100px', 'min-width': '75px', 'max-width': '100px' },
21             type: 'text'
22         }
23     ];
24     return columns;
25 }
26
27 public getCovid19Data() {
28     return this.http.get<Covid19Data>('https://api.covid19india.org/data.json');
29 }
30
31 public getDistrictWiseData() {
32     return this.http.get<DistrictWiseData>('https://api.covid19india.org/state_district_wise.json');
33 }

```

Figure 22 dataservice.service.ts

```

1  export interface Covid19Data {
2      cases time series: Array<CasesData>;
3      statewise: Array<StateWiseCases>;
4      tested: Array<TestingData>;
5  }
6
7  export interface CasesData {
8      dailyconfirmed: string;
9      dailydeceased: string;
10     dailyrecovered: string;
11     date: string;
12     totalconfirmed: string;
13     totaldeceased: string;
14     totalrecovered: string;
15 }
16
17 export interface StateWiseCases {
18     active: string;
19     confirmed: string;
20     deaths: string;
21     deltaconfirmed: string;
22     deltadeaths: string;
23     deltarecovered: string;
24     lastupdatedtime: string;
25     recovered: string;
26     state: string;
27     statecode: string;
28 }
29
30 export interface TestingData {
31     individualstestedperconfirmedcase: string;
32     positivecasesfromsamplesreported: string;
33 }

```

Figure 23 data-interface.ts

```

1  import { Component, Input, OnInit } from '@angular/core';
2
3  @Component({
4      selector: 'app-dashboard-card',
5      templateUrl: './dashboard-card.component.html',
6      styleUrls: ['./dashboard-card.component.css']
7  })
8  export class DashboardCardComponent implements OnInit {
9
10     @Input('Confirmed')
11     Confirmed:any;
12     @Input('Active')
13     Active:any;
14     @Input('Death')
15     Death:any;
16     @Input('Recovered')
17     Recovered:any;
18     @Input('dailyc')
19     DailyConfirmed:any;
20     @Input('dailyd')
21     DailyDeceased:any;
22     @Input('dailyr')
23     DailyRecovered:any;
24
25     constructor() { }
26
27     ngOnInit(): void {
28     }
29 }

```

Figure 24 dashboard-card.component.html

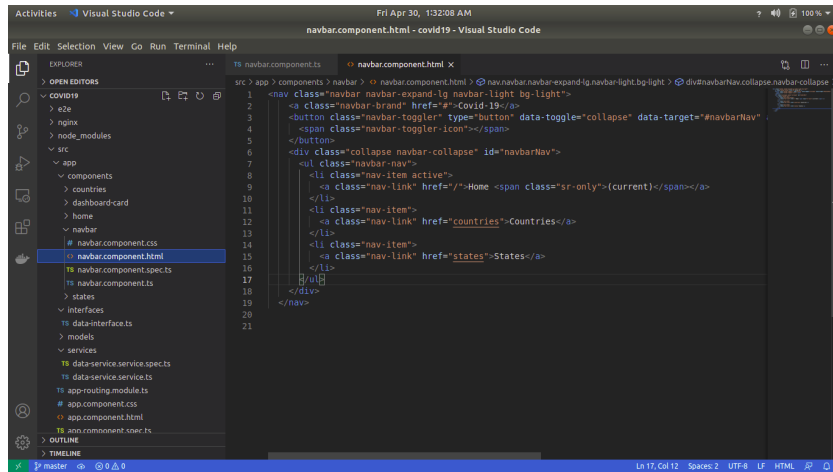


Figure 25 navbar.component.html

Passing the value of cases variable to the dashboard components in Indian format

```
public getDailyData():void{
  this.dataservice.getCovid19Data().subscribe((data:Covid19Data)=>{
    this.covid19Data = data.cases_time_series;
    for(const[key,value]of Object.entries(this.covid19Data)){
      var x:any=value;
      this.DailyConfirmed=parseInt(x.dailyconfirmed);
      this.DailyDeceased=parseInt(x.dailydeceased);
      this.DailyRecovered=parseInt(x.dailyrecovered);
    }
    // converting number to string and this is value passed to dashboard component
    this.DailyConfirmed=this.DailyConfirmed.toLocaleString('en-IN', {maximumFractionDigits:2});
    this.DailyDeceased=this.DailyDeceased.toLocaleString('en-IN', {maximumFractionDigits:2});
    this.DailyRecovered=this.DailyRecovered.toLocaleString('en-IN', {maximumFractionDigits:2});
  });
}
```

It will fetch the state wise data

```
transformDistrictWiseData(districtWiseData:any): void {
  for (const [key, value] of Object.entries(districtWiseData)) {
    var x:any=value
    for (const [key1, value1] of Object.entries(x)) {
      var y:any=value1
      for (const [key2, value2] of Object.entries(y)) {
        var z:any=value2;
        if (key1 !== 'statecode') {
          this.totalConfirmed=this.totalConfirmed+z.confirmed;
          this.totalDeath+=z['deceased'];
          this.totalRecovered+=z['recovered'];
          this.totalActive+=z['active'];
          this.transformedDistrictWiseData.push({ state: key, district: key2, confirmed: z['confirmed'],active:z['active'],recovered:z[
        ]
      }
    }
  }
}
```

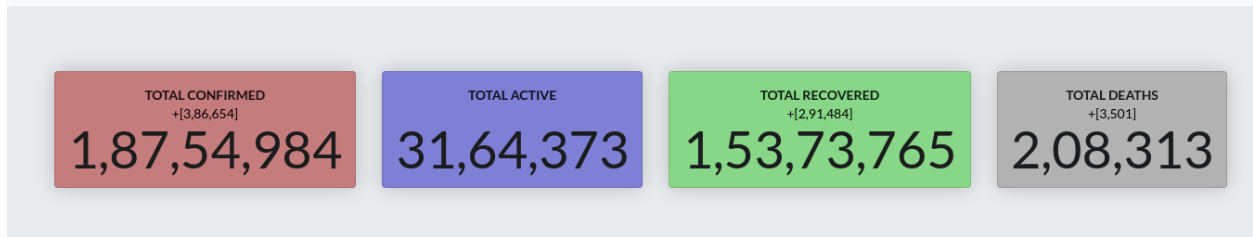


Figure 26 Dashboard-card

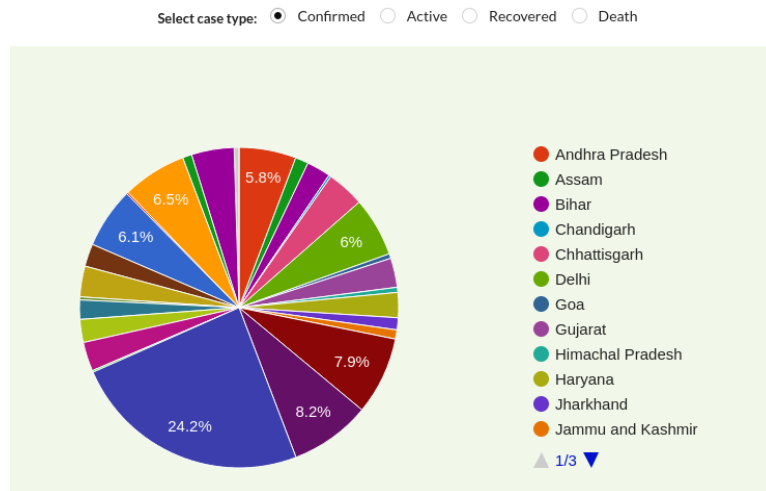


Figure 27 Pie chart of confirmed cases

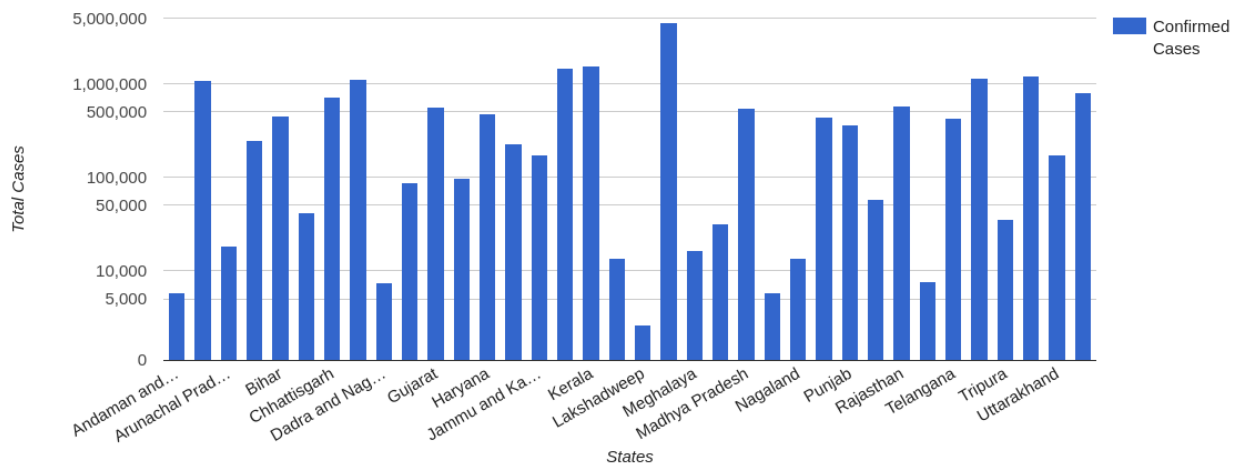


Figure 28 Barcharts of confirmed cases

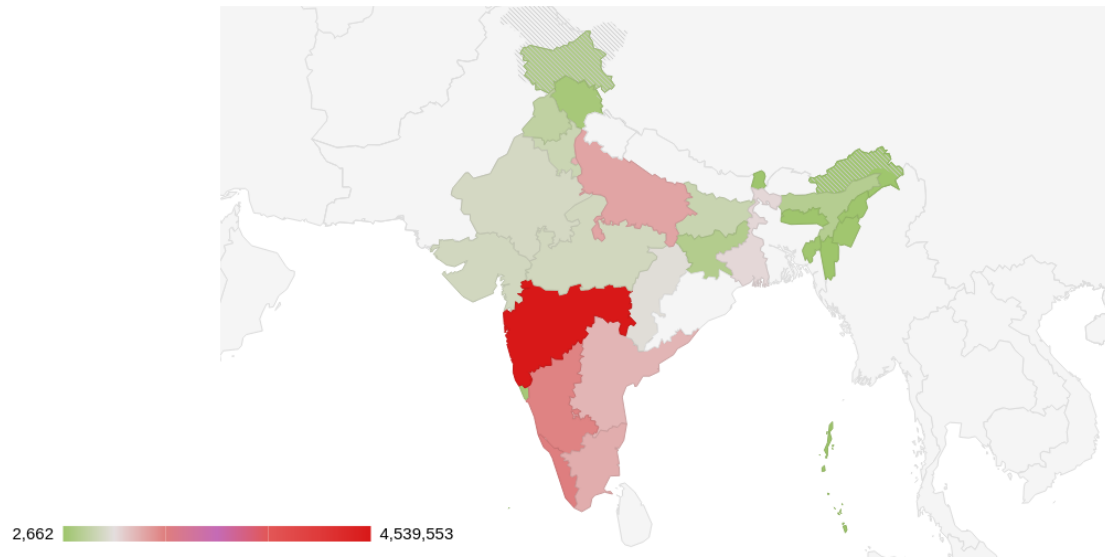


Figure 29 Map of Confirmed cases

STATE DATA				
STATE/UT	CONFIRMED	ACTIVE	RECOVERED	DEATHS
Maharashtra	4539553	670301	3799266	67985
Kerala	1533985	284085	1244301	5260
Karnataka	1474846	349496	1110025	15306
Andhra Pradesh	1084336	114158	962250	7928
Tamil Nadu	1148064	112556	1021575	13933
Delhi	1122286	97977	1008537	15772
Uttar Pradesh	1217952	309237	896477	12238
West Bengal	810955	110241	689466	11248
Odisha	435513	53031	380400	2082
Rajasthan	580846	169519	407243	4084
Chhattisgarh	713706	117910	587484	8312
Telangana	427960	76060	349692	2208

Figure 30 State wise cases

STATE/UT	CONFIRMED	ACTIVE	RECOVERED	DEATHS
Maharashtra	4539553	670301	3799266	67985
DISTRICT	CONFIRMED	ACTIVE	RECOVERED	DECEASED
Ahmednagar	167199	23124	142124	1950
Akola	40186	4741	34831	610
Amravati	63299	7243	55184	870
Aurangabad	123533	14649	106925	1945
Beed	53470	10788	41795	878
Bhandara	49737	11309	37961	463
Buldhana	44339	7927	36046	361
Chandrapur	60016	26613	32784	617
Dhule	37535	4050	33042	433
Gadchiroli	19421	4214	15014	184
Gondia	32054	7311	24404	333
Hingoli	13363	1935	11254	174
Jalgaon	117416	13393	102108	1887
Jalna	43713	6848	36233	631
Kolhapur	64809	7803	55215	1788
Latur	70413	12526	56788	1095
Mumbai	644583	67255	562738	13036
Mumbai Suburban	0	0	0	0
Nagpur	413981	80028	328817	5090
Nanded	80628	9360	69682	1578
Nandurbar	34491	7377	26571	542
Nashik	308118	52954	252129	3034
Osmanabad	38456	7606	29957	875
Other State	146	26	0	118
Palghar	86961	16759	68967	1225
Parbhani	35371	10059	24733	568
Pune	833243	104529	719221	9438

Figure 31 District wise cases

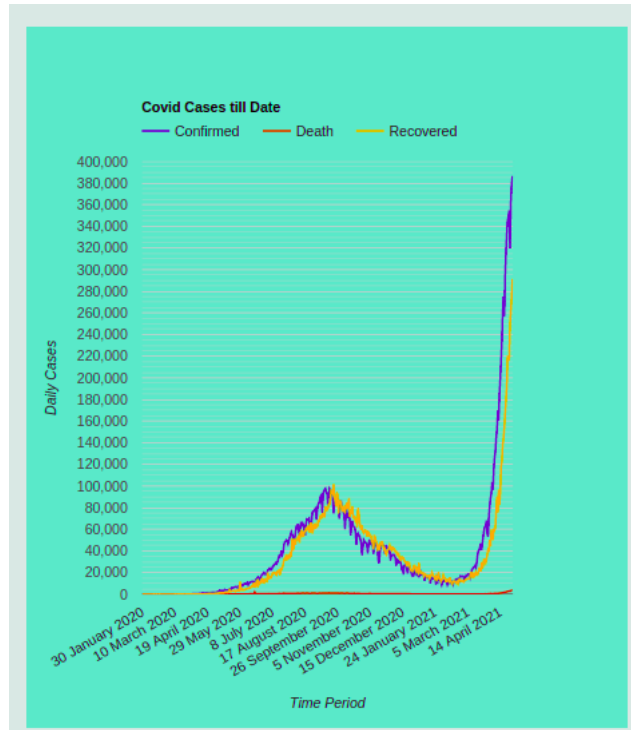


Figure 32 cases till date

## 6. CONCLUSION AND FUTURE WORK

We have showcased the web application that plots Covid19 time series data using a live data source . It allow user to select the state and view the district data .However more functionality related to state like showing the graph when clicking on the state, adding more information on the column table other than cases like population , number of testing per millions. Later on if public api for vaccination data is available than based on this information more things can be drawn

### Here are the key takeaways of this Project

- Working with an HTTP request
- Working with JSON Data (Arrays and Objects)
- Sorting Numeric Data
- Sorting Datatypes.
- Sorting String and Object arrays
- Doing arithmetic Calculation on Object types
- Nesting Table inside Template
- Loading component Dymanically on Click
- Cross Component communication

## REFERENCES

- <https://www.npmjs.com/package/ng2-google-charts>
- <https://angular.io/docs>
- <https://medium.com/@kamalaujla4vr/display-data-on-indias-geo-chart-in-angular-8-332e6f1b01dd>

- <https://documenter.getpostman.com/view/10724784/SzYXXKmA?version=latest>

## LINKS

**Github Repo** → <https://github.com/njetha/Covid19-Data-Visualization>

**DockerHub Repo** → <https://hub.docker.com/repository/docker/njetha/covid19>