A
Minor Project-1 Report
On

"COVID VISUALIZER"

Submitted in partial fulfilment of
The requirements for the 5th Semester Sessional Examination of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

By

NAME
Abhisek Ray, Satyam Kumar, Purushottam Kumar
Registration No.
1901060224, 1901060353, 1901060293

Under the able Supervision of

Ms Priyanka Priyadarsini

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GIET UNIVERSITY, Gunupur

2021 - 22

GIET UNIVERSITY,GUNUPUR

Dist. - Rayagada, Odisha-765022, Contact:- +91 7735745535,
06857-250170,172, Visit us:- www.giet.edu

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "COVID VISUALIZER" is done by Name- Abhisek Ray, Satyam Kumar, Purushottam Kumar, Regd. No.- 1901060224, 1901060353 , 1901060293 in partial fulfilment of the requirements for the 5th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2021-22. This work is submitted to the department as a part of an evaluation of the 5th Semester Minor Project-1.

Project Supervisor *Class Teacher*

Mr.Bhavani Sankar Panda
Project Coordinator, 3rd Year

Dr .Sanjay Kumar Kuanar
HoD, CSE

Acknowledgement

I would like to thank my supervisor ms. Priyanka Priyadarshini for support and doubt clearing at any moment when we face trouble in our project.

Secondly, I would like to thank my class teacher Gitanjali Mishra for their good explanation about how to add some mind-blowing features.

Thirdly I would like to thank our Project Coordinator Mr Bhavani Sankar Panda for giving a good idea about the project.

Special gratitude to our HOD Dr Sanjay Kumar Kuanar, for giving information about the topic.

Name of Students

Abhisek Ray
Satyam Kumar
Purushottam Kumar

Abstract

Since December 2019 the world is experiencing a deadly disease caused by a novel coronavirus termed as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The disease associated with this virus is known as COVID-19. This paper focuses on COVID-19 based on freely available data including the ones in the Github repository.

Data analytics is provided on a number of aspects of COVID-19 includes the symptoms of this disease, the difference of COVID-19 with other diseases caused by severe acute respiratory syndrome (SARS), Middle East respiratory syndrome (MERS), and swine flu. The impact of temperature on the spread of COVID-19 is also discussed based on the website. Moreover,

data visualization is provided on the comparison of infections in different countries in the world and even also global wise data. In another route, we show only individual Covid updates in India. which shows that males are more prone to this disease and older people are more at risk.

Based on the data, the pattern in the increase of confirmed cases is found to be an exponential curve in nature. Finally, the relative number of confirmed, recovered and death cases in different countries are shown with data visualization.

Then discuss some benefits, building blocks, and useful resources which might help for future exposure.
We always discover many things on many social media platforms, But here our motive is to combine all things get-together in various aspects.

Covid Visualizer is Fullstack based WebAPP. In this project, we used Gatsby.js, Chat.js, D3.js, and Leaflet.js to add some stunning frontend and Visualization features. And we combined Firebase User Services to Provide Scalable and robust Infrastructure as well as User Authentications. By the use of Nodejs, we enabled development faster along with MongoDB allows to store, retrieve data that allows developers to move fast and handle huge volumes of data and traffic.

The main objective of building this project is to provide a dashboard to users, where user can view their data by the means of visualizations, maps, corona cases in the world and with the mobile responsive View. User can control their profile as well. It is intuitive, requires minimum training, and comes with automation features that boost productivity.

The scope is lesser hardware costs, High-quality software, Integrated Management, scaling and source code is Opensource to everyone.it enables typically offering multiple ways to solve problems also opens business opportunities.

Introduction

Purpose

This Report analyzes COVID-19 based on the currently available data [12,20]. Analytics is provided on several aspects of COVID-19 including the symptoms, the difference with other viruses, and the impact of temperature on over the world.

Moreover, data visualization is provided on the comparison of infections in different countries, and the pattern in the increase of confirmed cases and the relative number of confirmed /recovered/death cases in different countries. The rest of the report is organized as follows. Section 2 describes the different aspects of COVID-19 using tabular data. Section 3 provides visualization of how the infection has spread across the world using pie charts and bar charts. Section 4 provides concluding remarks.

Project Scope

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application,

and it will work for a particular user or
organization. Later the project can be modified to operate smoothly and add new features to become more user friendly.

In our project there are three kinds of data being shown on the web one is chart-based, one is table-based and another is pie-chat based.
This kind of may be easy for the user to understand how much the country is affected by the pandemic.

The world cases are showing you the entire world data on one surface with a graph developed using chart.js and with their infected, active, death and recovered cases. In that graph, the graph is denoted with the dates.

In the very future, we may add a new feature i.e Feed from the user which is related to their post on the Covid-visualizer app section. There are support, comment and share option are available.

There are only two types of users that could present one is verified and another one is spam. The verified is the user who has correct user data and spam has no verified tag because may he/she doesn't have to attend any survey related to the web app.

Basically verified users can be trusted in any condition but spam user can not.

In that user based login. We may give you to handle your user data in real time. If any user wanted to post a message in the feed of the web app. Show they can choose the message related to whether a suffer or a supporter.

Product Features

By Developing Graphical content on the web, we can show the live updates of Covid-19 cases in particular locations and their states.
Handle Frequently, Consistently, relevant information for your audience in real-time without any buffer.
Post things that get your audience excited Start with a plan, and you'll always have something to post!
By Some leading Authentication features, user can use their google and manual email account to create an Account.
This project has some Good UI effects as well have a better user management system, by which this web app can stand out.

The main objective to make this COVID VISUALIZER is to take some control over the audience to know the insight of the society as a whole, what they find interesting or many more.
We are going through from the very beginning to know some technical Features of this Covid visualizer application.

Improved development process = cost and time benefit to our clients

Improved performance = faster applications

Improved SEO = more indexable, SEO friendly applications

System Analysis

Hardware Requirements

Even at least 4GB ram is convenient(not using any virtual emulator), we suggest 8GB or 16 rams(if you are using Android Studio or Xcode like emulator software). Have a suitable system like windows, Linux, macOS and also have a 4-6 cores processor, which may avoid the lagging in your development.

Software Requirement

Have a good Code editor like VsCode and IntelliJ IDEA
installed in your system.
Required Node.js run-time Environment to be installed.
If you have any running browser in your system, it will work fine. But Chrome has some advance and stable support. so you can even switch.

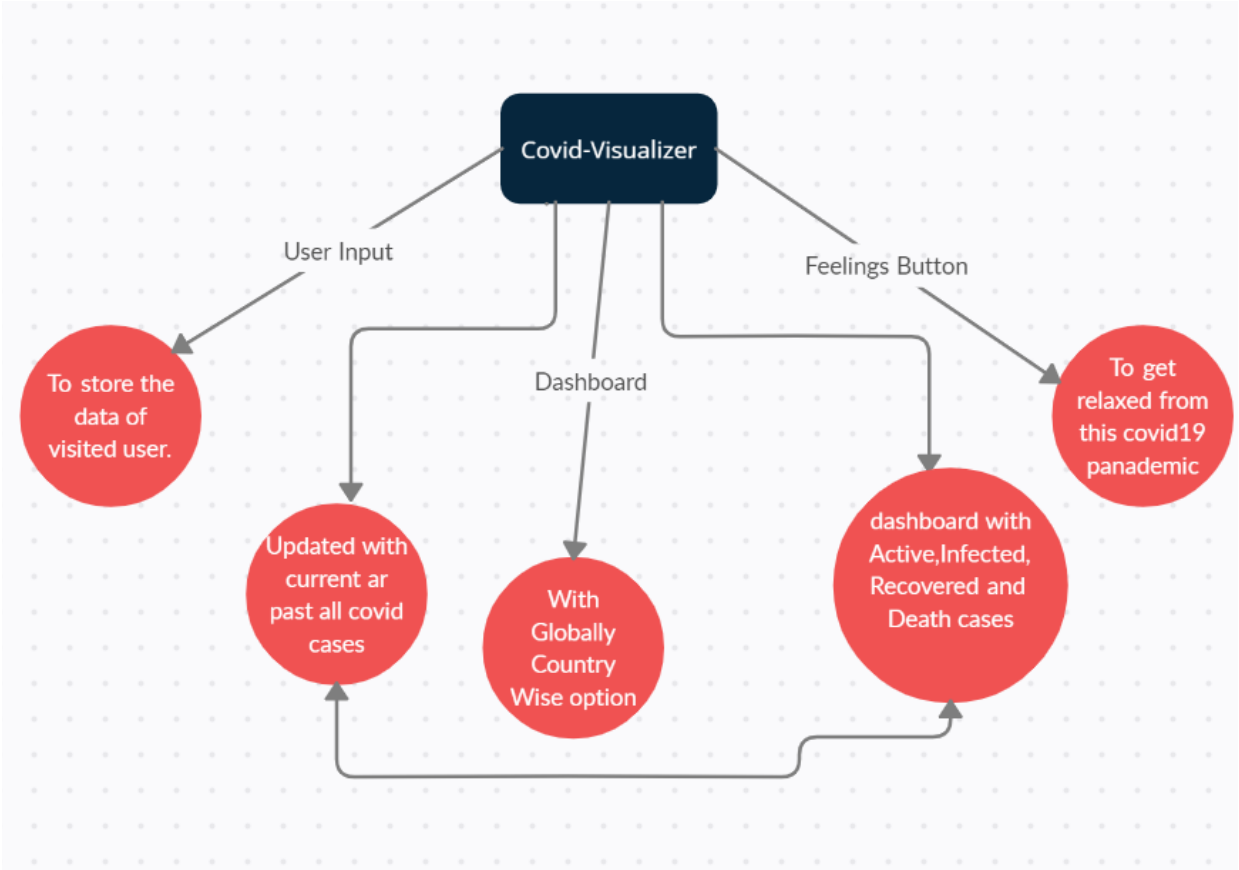
We use MongoDB Atlas for Authentication and for our Ui
used Gatsby.js, chart.js, Leaflet.js, D3.js

And for Google authentication, we used Google Firebase.

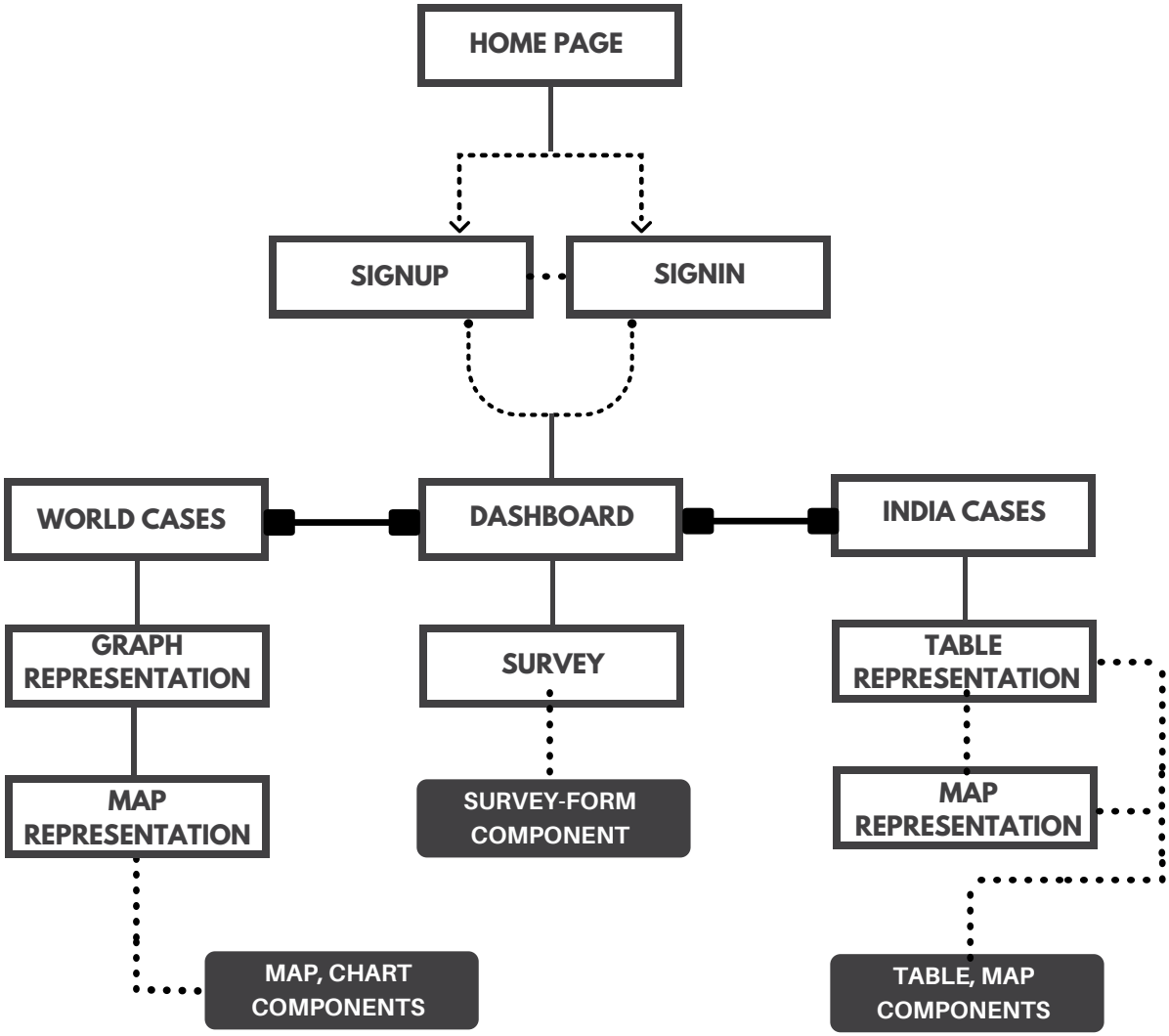
System Design & Specifications

High Level Design (HLD)

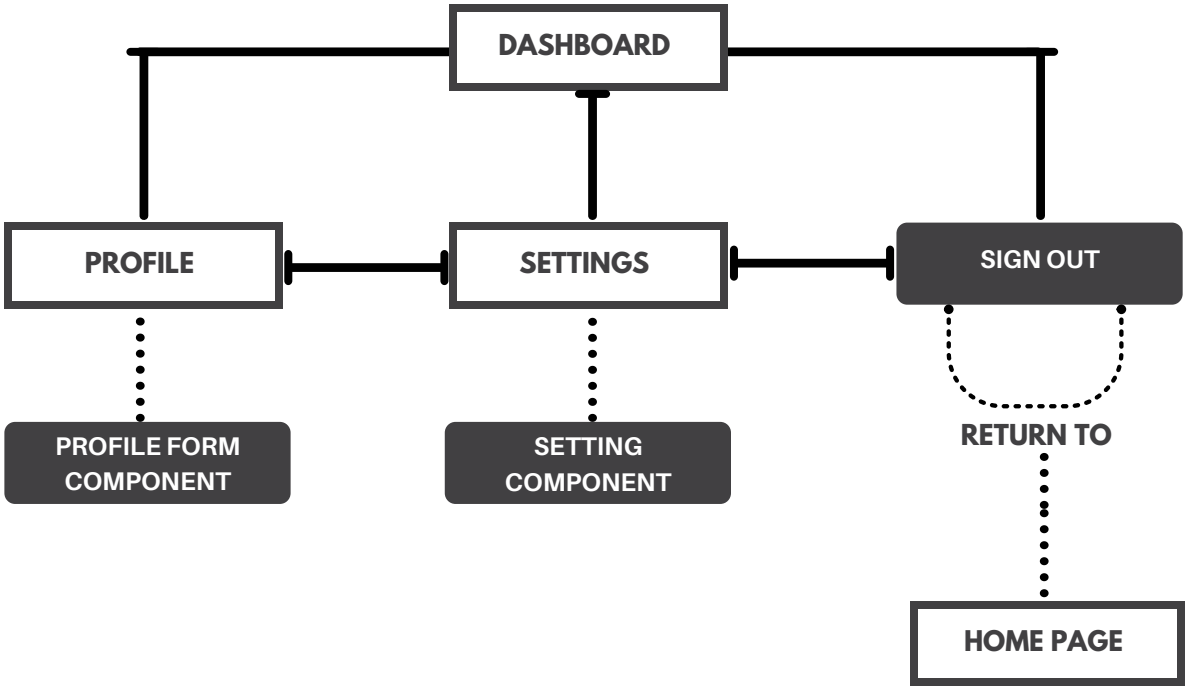
Flow chart



E-R Diagram



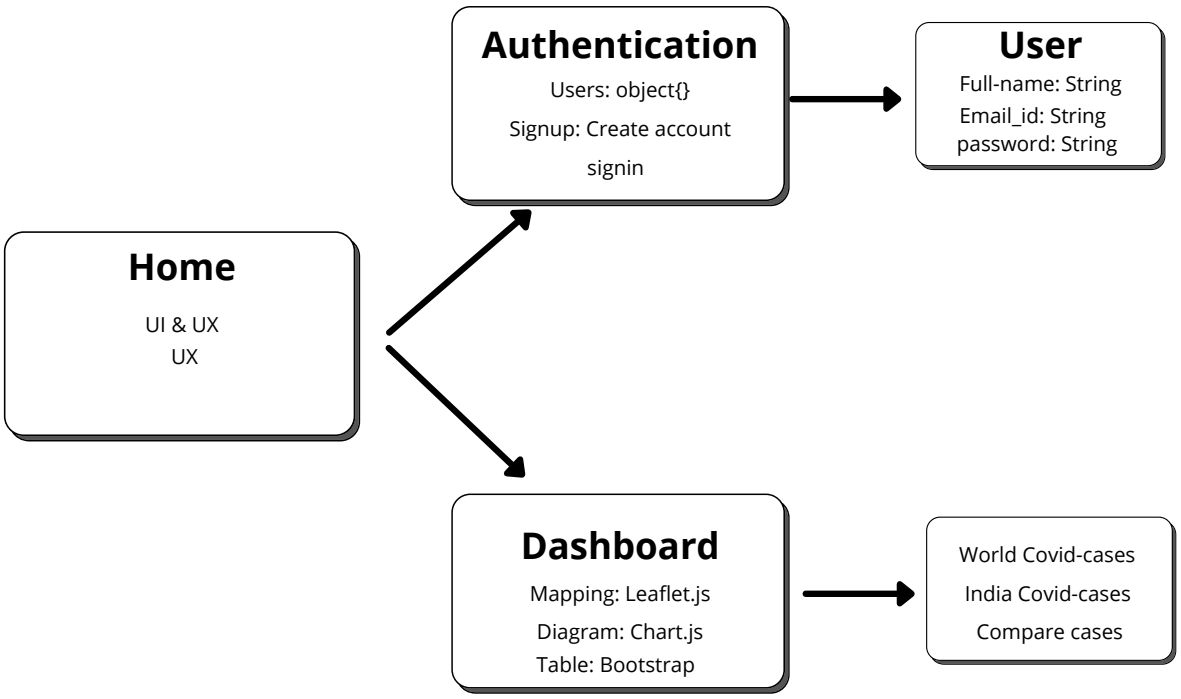
Overall project ER-diagram



User Profile ER-diagram

Low Level Design (LLD)

Process Specification (Pseudo code / Algorithm)



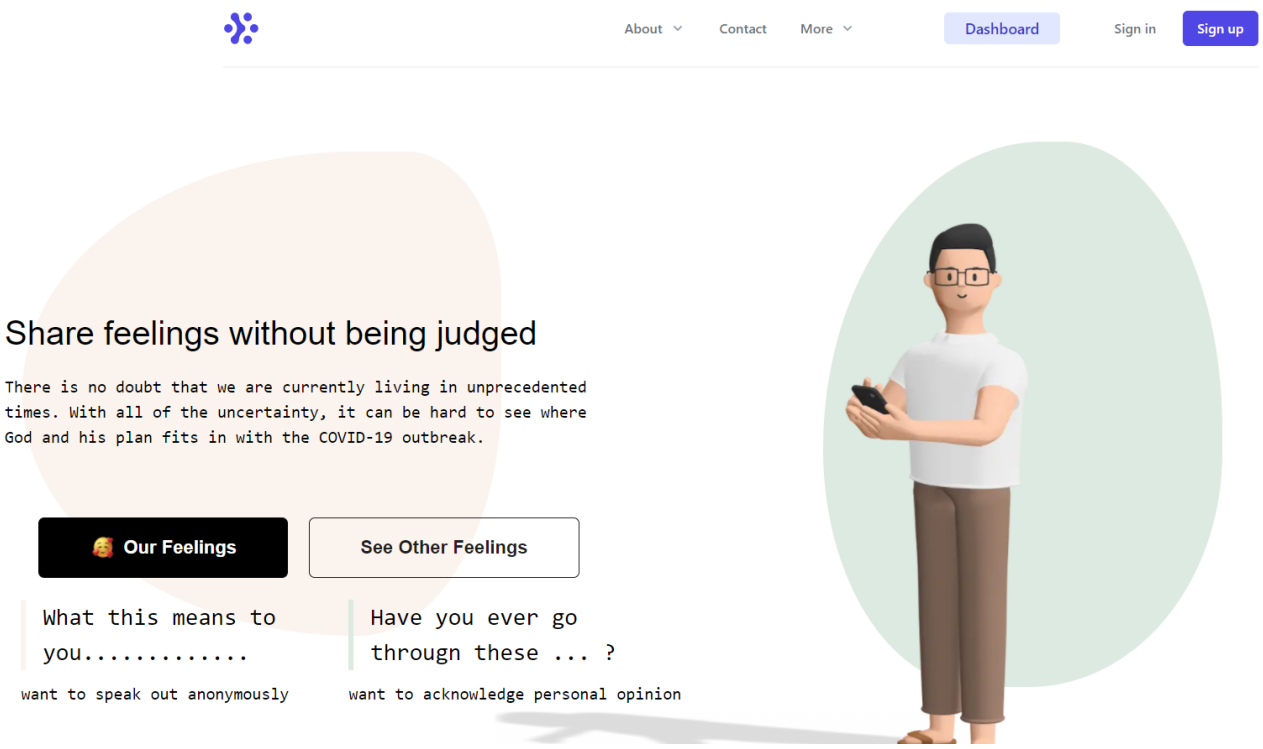
There are only two types of users that could present one is verified and another one is spam. The verified is the user who has correct user data and spam has no verified tag because may he/she doesn't have to attend any survey related to the web app.

Basically verified users can be trusted in any condition but spam user can not.

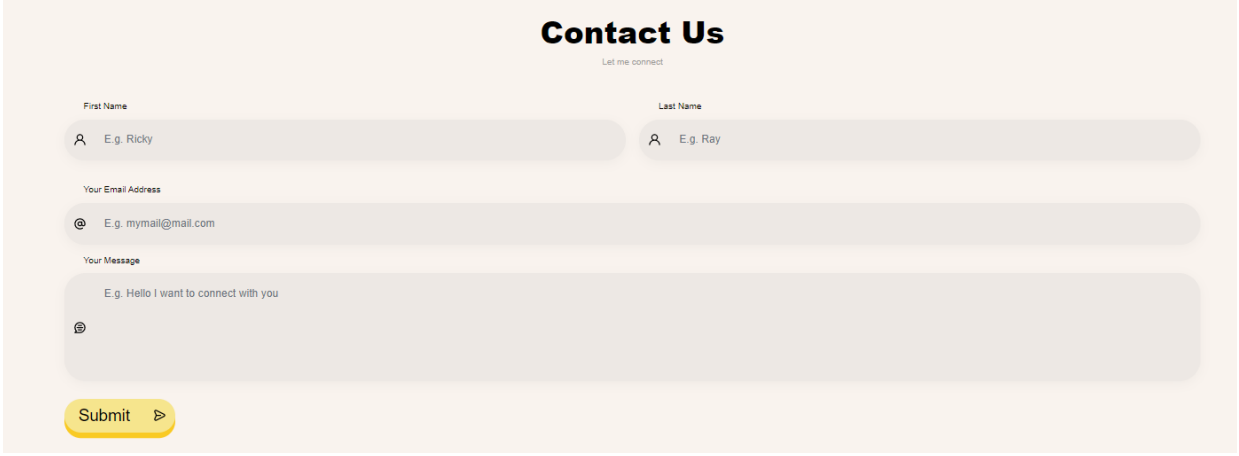
And here the above is showing the LLD of our Website.

Screen-Shot Diagram

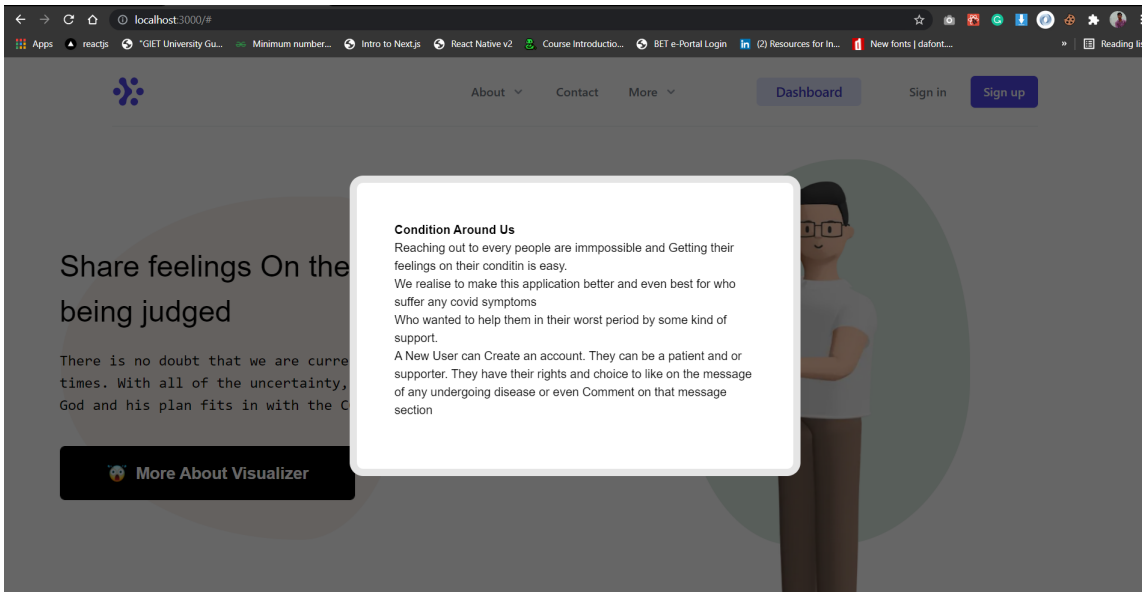
First Part of Homepage



Contact Part of Homepage

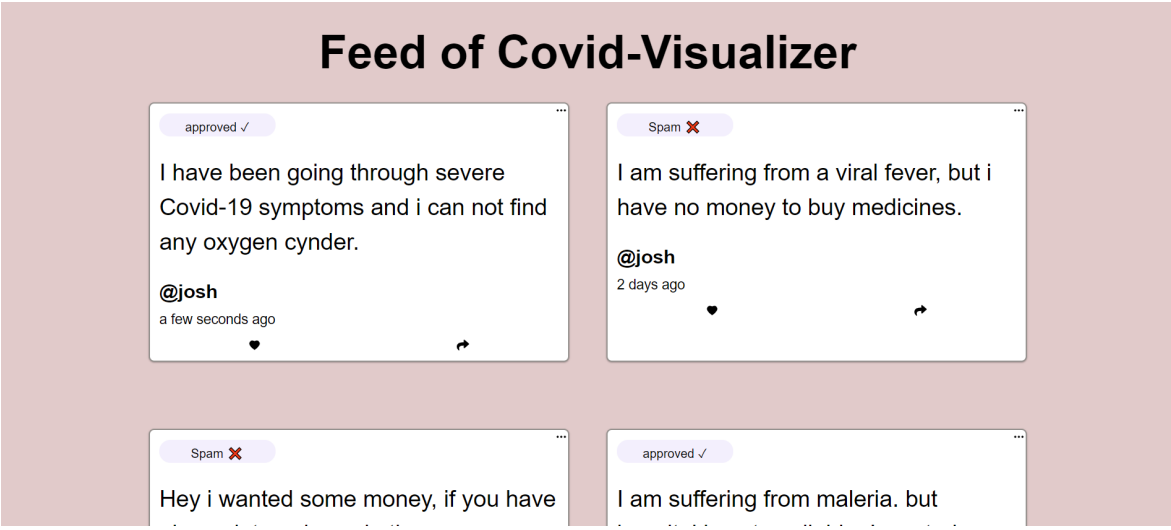


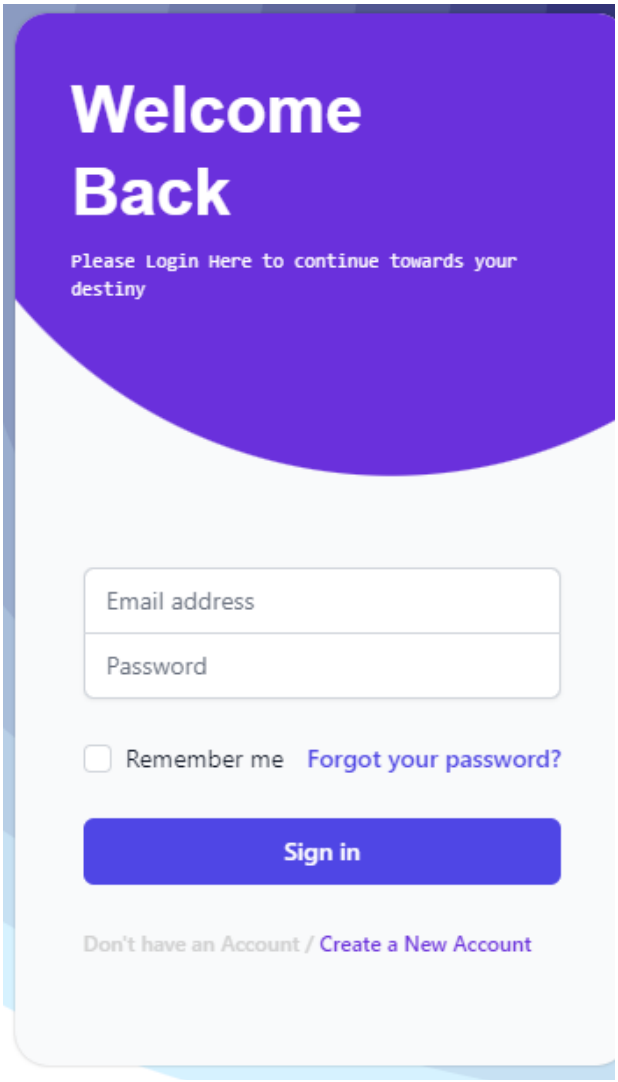
Popup to See More about us

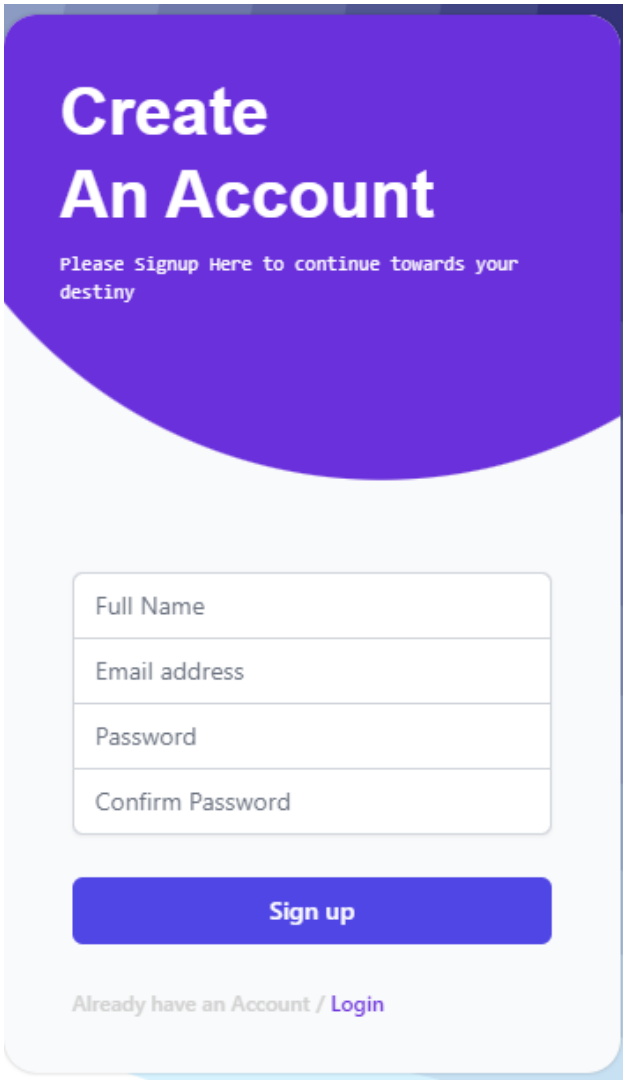


About Our Website

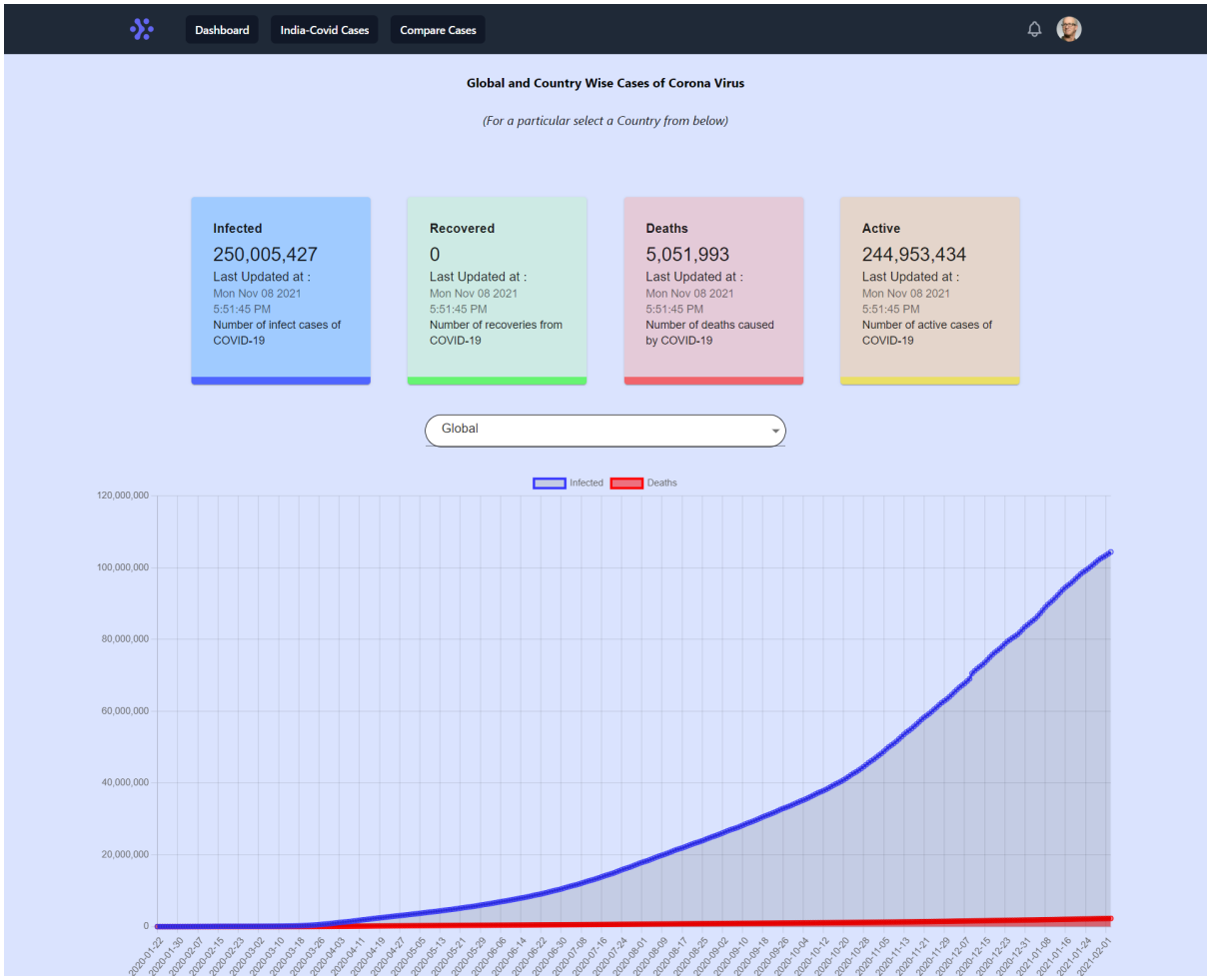
Feed of Covid Visuallization





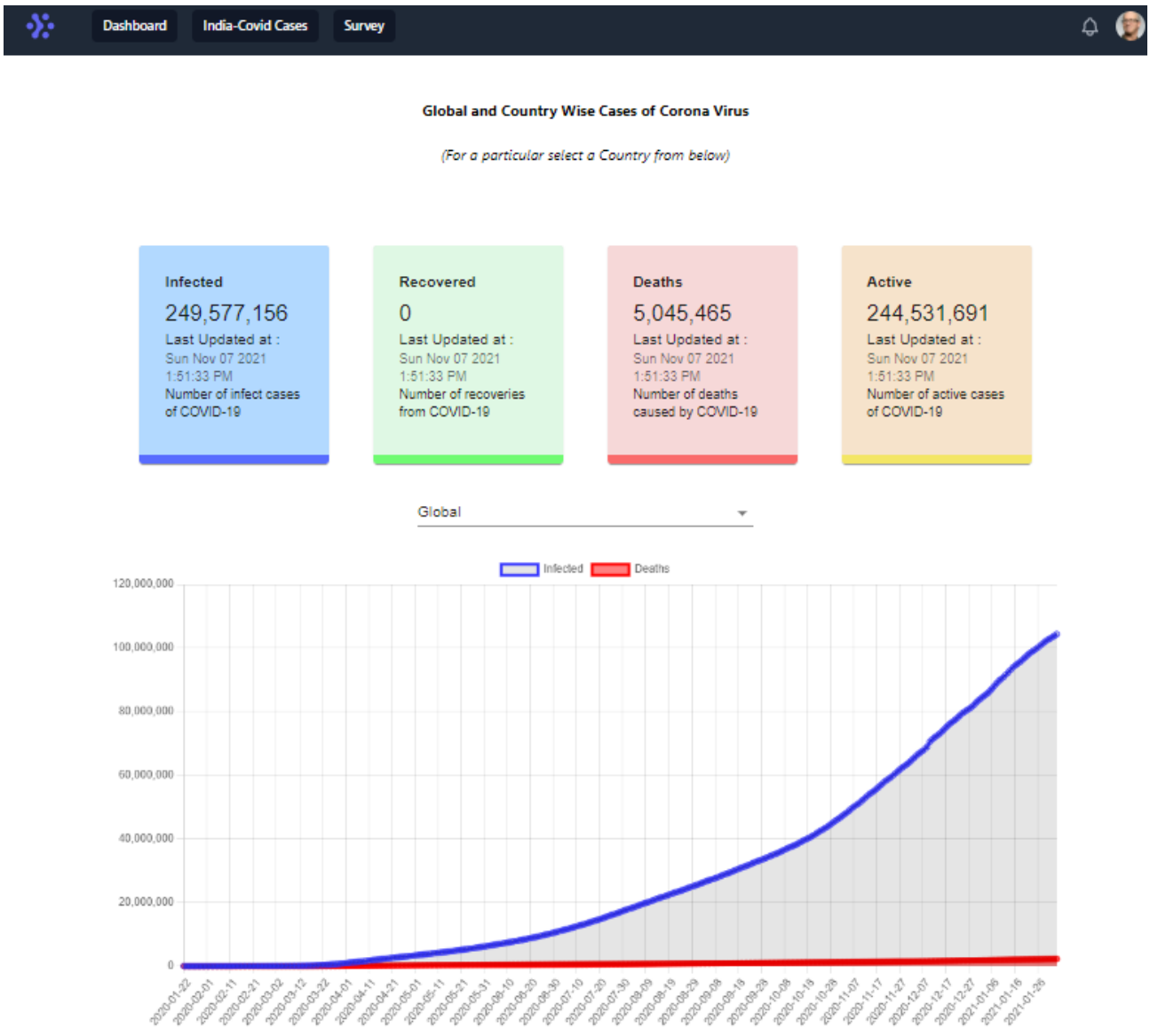


Here we comes with Sign-in and sign-up option

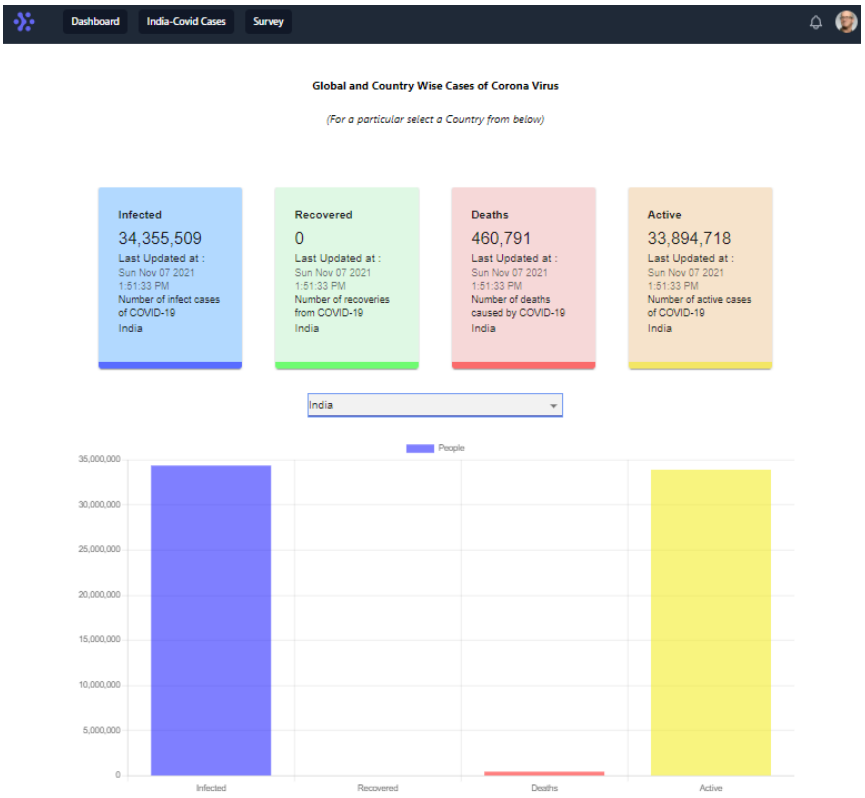


Dashboard of Covid-visulizer

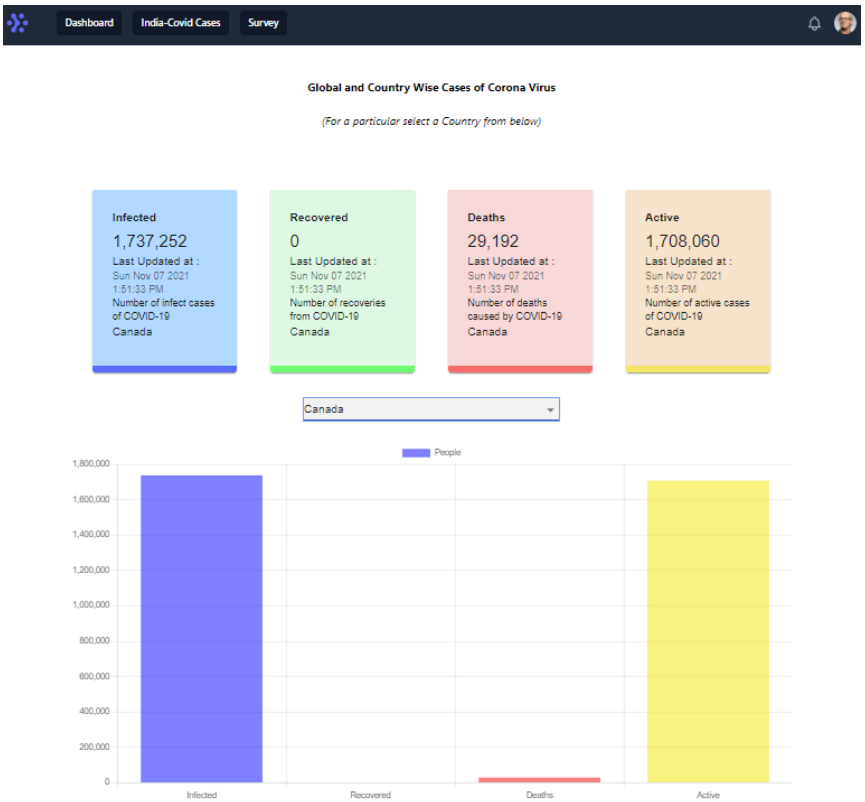
Globally Dashboard with Infected,Recovered, Deaths and Active Cases



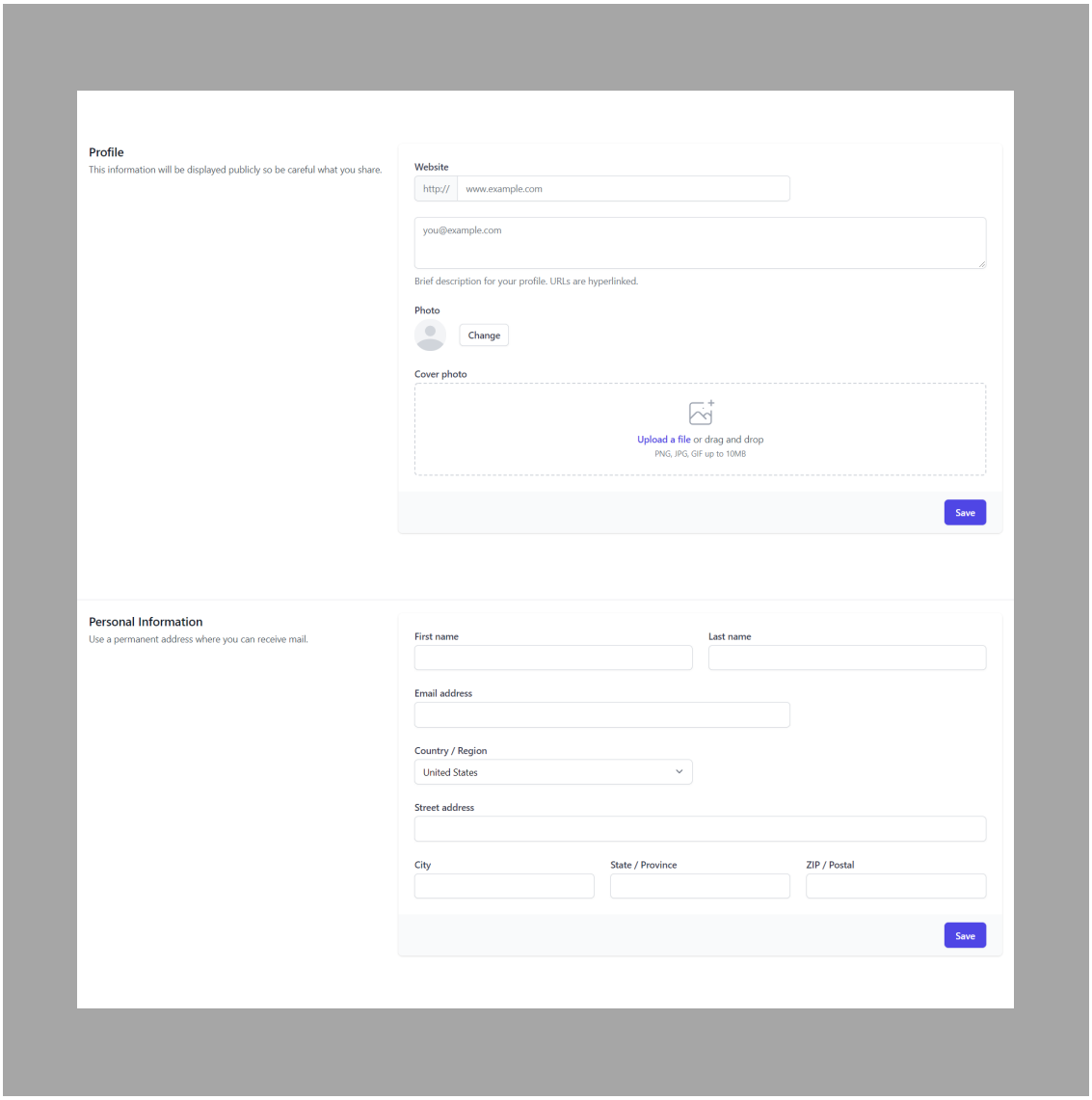
Country Wise Dashboard with Infected,Recovered,Deaths and Active Cases



India



Canada



Setting Page

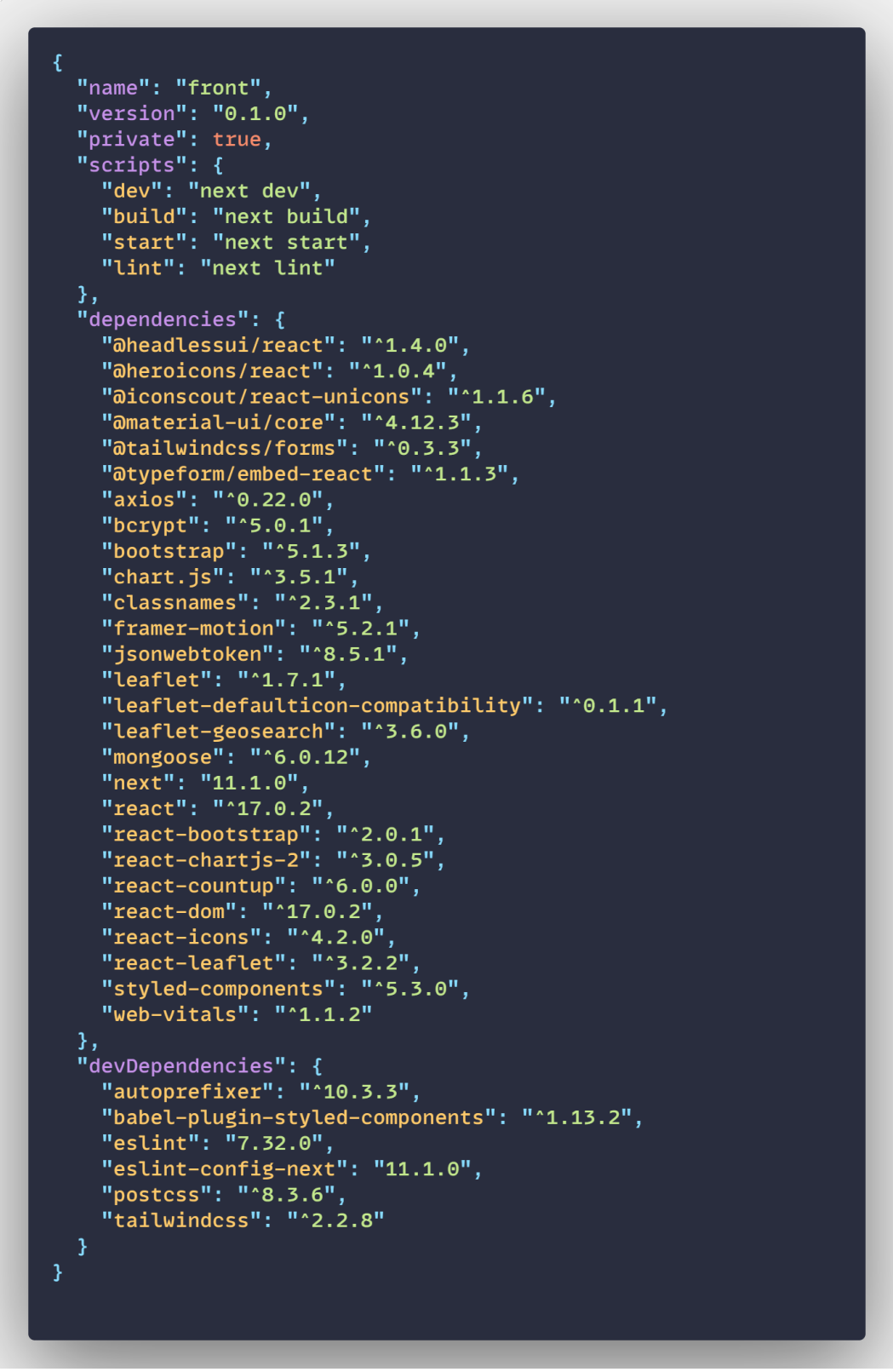
Coding

We use git version control to communicate with other codes
in our project group. Git hub even helps to manage our work easier and even more fun by which in real-time we can share our code and any updates related to the project.

This below is our Repository link to our project
https://github.com/Abhisek-Ray99/nextjs-5th-minor

Without Github, manually sharing the code with your team member is a really difficult task. So this is how we manage to write code with gatsby.js and for styling, we use styled component and CSS module.

Rather than that, we use chart.js, leaflet.js and another library like tailwind for styling our components.



```
{
  "name": "front",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "@headlessui/react": "^1.4.0",
    "@heroicons/react": "^1.0.4",
    "@iconscout/react-unicons": "^1.1.6",
    "@material-ui/core": "^4.12.3",
    "@tailwindcss/forms": "^0.3.3",
    "@typeform/embed-react": "^1.1.3",
    "axios": "^0.22.0",
    "bcrypt": "^5.0.1",
    "bootstrap": "^5.1.3",
    "chart.js": "^3.5.1",
    "classnames": "^2.3.1",
    "framer-motion": "^5.2.1",
    "jsonwebtoken": "^8.5.1",
    "leaflet": "^1.7.1",
    "leaflet-defaulticon-compatibility": "^0.1.1",
    "leaflet-geosearch": "^3.6.0",
    "mongoose": "^6.0.12",
    "next": "11.1.0",
    "react": "^17.0.2",
    "react-bootstrap": "^2.0.1",
    "react-chartjs-2": "^3.0.5",
    "react-countup": "^6.0.0",
    "react-dom": "^17.0.2",
    "react-icons": "^4.2.0",
    "react-leaflet": "^3.2.2",
    "styled-components": "^5.3.0",
    "web-vitals": "^1.1.2"
  },
  "devDependencies": {
    "autoprefixer": "^10.3.3",
    "babel-plugin-styled-components": "^1.13.2",
    "eslint": "7.32.0",
    "eslint-config-next": "11.1.0",
    "postcss": "^8.3.6",
    "tailwindcss": "^2.2.8"
  }
}
```

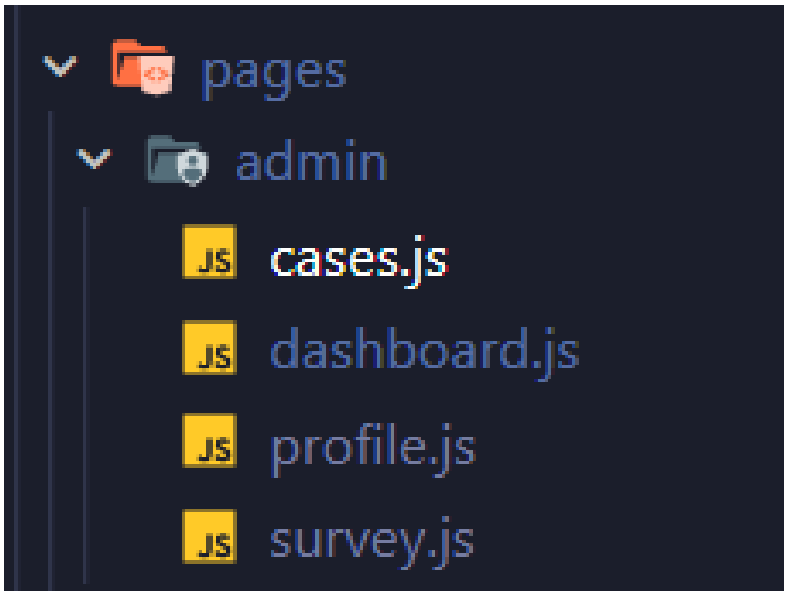
package.json

packages used in the project

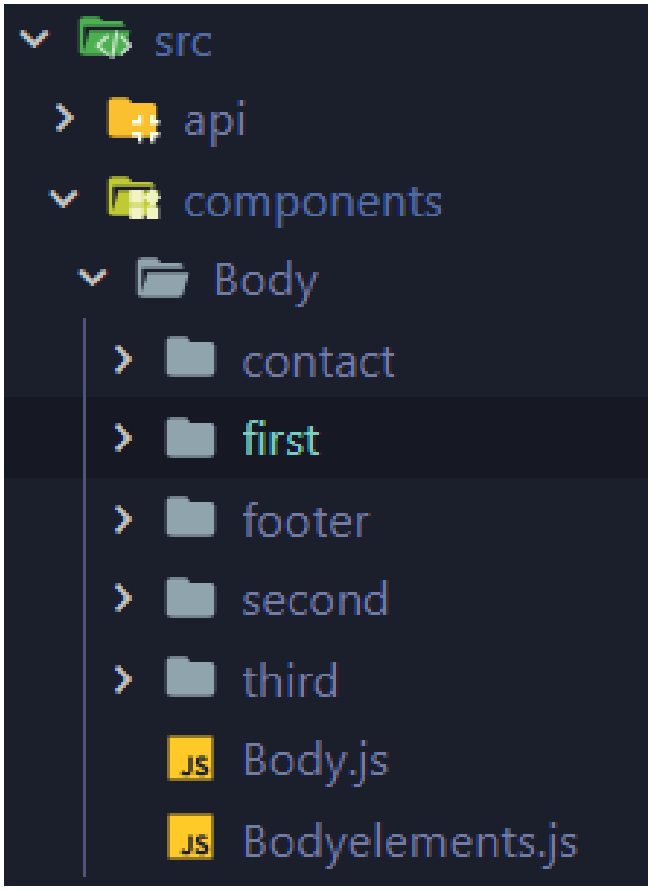
```jsx
import React from 'react'
import { Globalstyle } from './Globalstyles'
import styled from 'styled-components'

import Body from './components/Body/Body'
import Nav from './components/Nav/Nav'



const MyApp = styled.div`
    margin: 0;
    padding: 0;
`

const App = () => {
    return (
        <MyApp>
            <Globalstyle />
                <Nav />
                <Body />
        </MyApp>
    )
}

export default App;
```

App.js

Home page of Covid-Visualizer



Routes



Home page components

```jsx
import React from 'react'

import { Div } from './bodyelements';
import Contact from './contact';
import First from './first/Index';
import Footer from './footer';
import Second from './second/Index';
import Third from './third/Index';

const Body = () => {
    return (
        <Div>
            <First/>
            <Second/>
            <Third/>
            <Contact/>
            <Footer/>
        </Div>
    )
}

export default Body;
```

Home.js

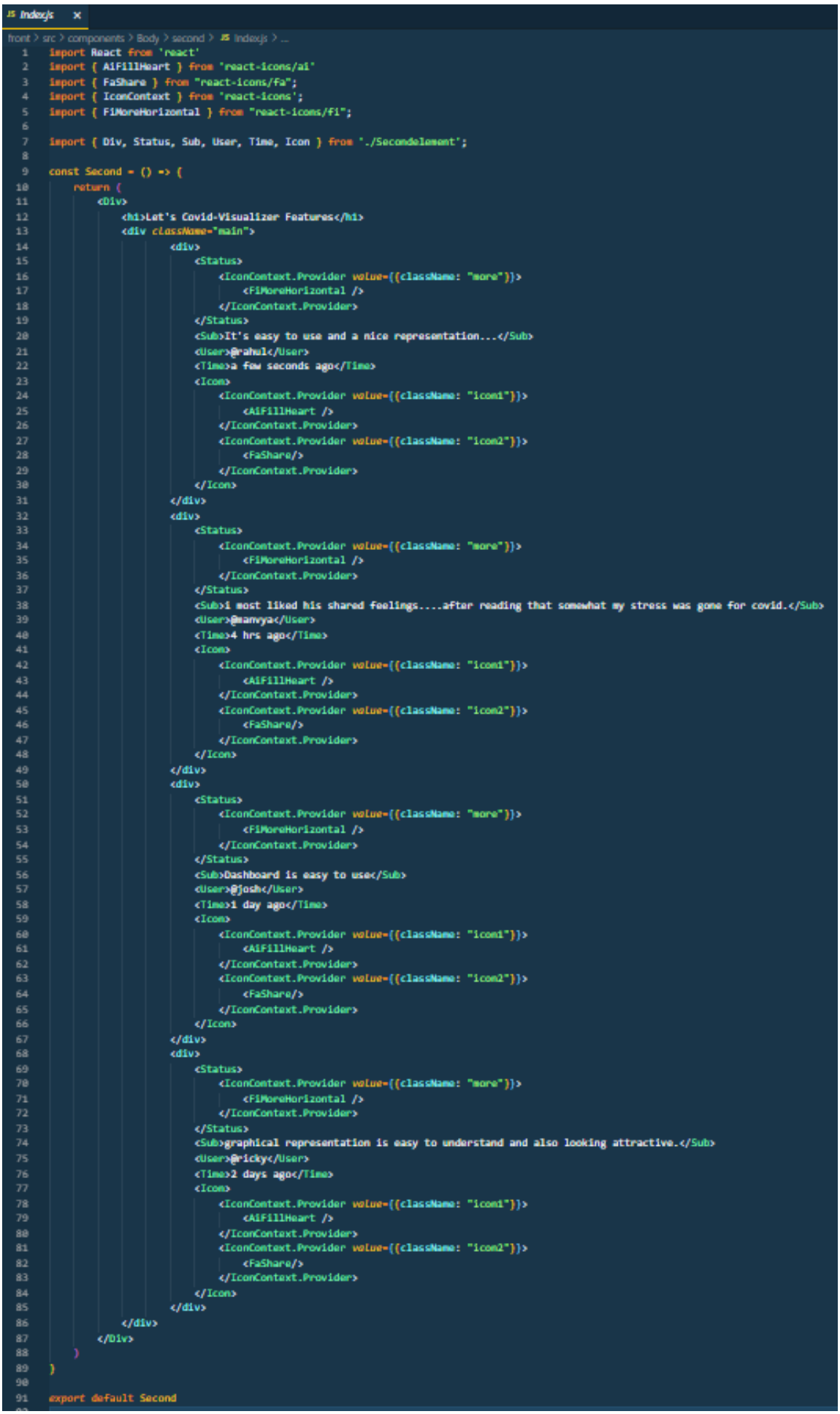
Contact Part of Homepage

First Part of Homepage

Index.js M

front > src > components > Body > first > Index.js > ...

```
import React from 'react';
import styled from 'styled-components';
import Link from 'next/link';
import Image from 'next/image'

import img2 from '../../../../public/img/7.png'
import img3 from '../../../../public/img/200w.gif'


import { Share, Div, Div_1, Head, Head_p, Div_2, Btn1, Btn2, Quote,StyledImage,Wrap } from './Firstelements'


const First = () => {
    return (
        <Div>
            <Share>
                <Div_1>
                    <Head>Share feelings without being judged</Head>
                    <Head_p>There is no doubt that we are currently living in unprecedented times. With all of the uncertainty,
                        it can be hard to see where God and his plan fits in with the COVID-19 outbreak.
                    <br/>

                    </Head_p>
                </Div_1>
                <Div_2>
                    <Btn1 href="/">
                        <>
                            <Image src={img3} />
                            <span>Our Feelings</span>
                        </>

                    </Btn1>
                    <Link href="/"><Btn2>See Other Feelings</Btn2></Link>
                </Div_2>
            </Share>
            <Quote>
                <div className="first-div">
                    <h1>What this means to you.............</h1>
                    <p>want to speak out anonymously</p>
                </div>
                <div className="second-div">
                    <h1>Have you ever go througn these ... ?</h1>
                    <p>want to acknowledge personal opinion</p>
                </div>
            </Quote>
            <Wrap>
                <Image
                    src={img2}
                    width={650}
                    height={650}
                />
            </Wrap>

        </Div>
    )
}

export default First

```



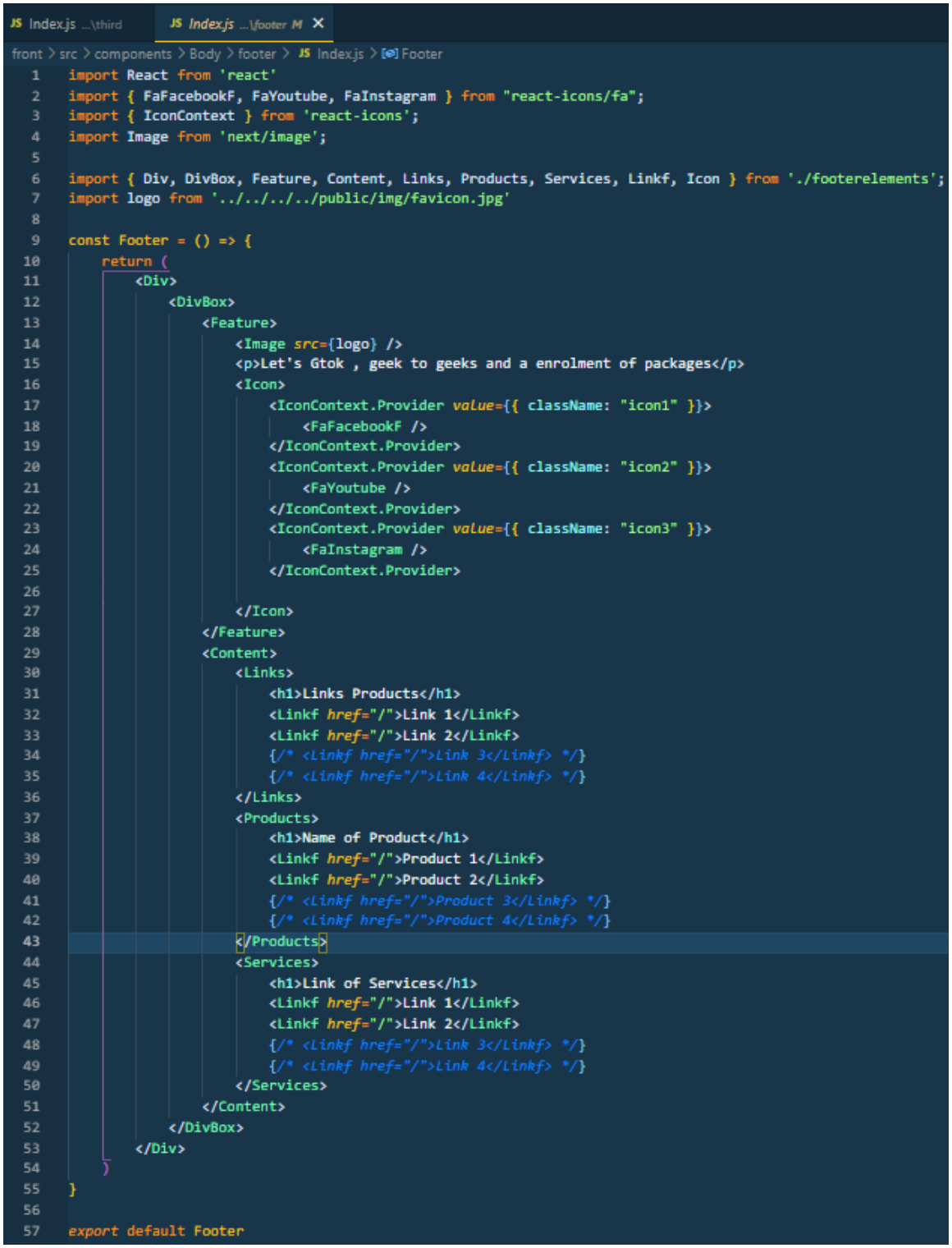
Second Part of Homepage

Third Part of Homepage

index.js M

front > src > components > Body > third > index.js > Third

```jsx
import React from 'react'
import { motion } from 'framer-motion';
import Image from 'next/image'

import { Section, Container, ColumnLeft, ColumnRight, Button, Img } from './Thirdelements';
import Frame1 from '../../../../public/img/3d1.png';
import Frame2 from '../../../../public/img/3d2.png';
import Frame3 from '../../../../public/img/3d3.png';
import Frame4 from '../../../../public/img/3d4.png';
import Frame5 from '../../../../public/img/3d5.png';
import Frame6 from '../../../../public/img/3d6.png';

const Third = () => {

    const fadeLeft = {
        hidden: { opacity: 0, x: -100 },
        visible: { opacity: 1, x: 0 }
    }

    return (
        <Section>
            <Container>
                <ColumnLeft>
                    <motion.h1
                        initial={{ opacity: 0 }}
                        animate={{ opacity: 1 }}
                        transition={{ duration: 1 }}
                    >Why let's Covid-Visualizer</motion.h1>
                    <motion.p
                        variants={fadeLeft}
                        initial='hidden'
                        animate='visible'
                        transition={{ duration: 1 }}
                    >It is no surprise that there is such a high volume of data relating to the recent COVID-19 outbreak.
                    Covid-Visualizer features high-performance graphical visualization to organize and display this data
                    in such a way that it can be easily understand by doctors, researchers, and even the public. </motion.p>
                    <Button
                        whileHover={{ scale: 1.05 }}
                        whileTap={{ scale: 0.95, backgroundColor: "#67F6E7" }}
                    >Learn More</Button>
                </ColumnLeft>
                <ColumnRight>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame1}/>
                    </Img>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame2} />
                    </Img>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame3} />
                    </Img>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame4} />
                    </Img>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame5} />
                    </Img>
                    <Img
                        whileTap={{ scale: 0.9 }}
                        drag={true}
                        dragConstraints={{ left: 0, right: 250, top: 0, bottom: 50 }}
                    >
                        <Image src={Frame6} />
                    </Img>
                </ColumnRight>
            </Container>
        </Section>
    )
}

export default Third
```



```
import React from 'react'
import { FaFacebookF, FaYoutube, FaInstagram } from "react-icons/fa";
import { IconContext } from 'react-icons';
import Image from 'next/image';

import { Div, DivBox, Feature, Content, Links, Products, Services, Linkf, Icon } from './footerelements';
import logo from '../../../../public/img/favicon.jpg'

const Footer = () => {
    return (
        <Div>
            <DivBox>
                <Feature>
                    <Image src={logo} />
                    <p>Let's Gtok , geek to geeks and a enrolment of packages</p>
                    <Icon>
                        <IconContext.Provider value={{ className: "icon1" }}>
                            <FaFacebookF />
                        </IconContext.Provider>
                        <IconContext.Provider value={{ className: "icon2" }}>
                            <FaYoutube />
                        </IconContext.Provider>
                        <IconContext.Provider value={{ className: "icon3" }}>
                            <FaInstagram />
                        </IconContext.Provider>

                    </Icon>
                </Feature>
                <Content>
                    <Links>
                        <h1>Links Products</h1>
                        <Linkf href="/">Link 1</Linkf>
                        <Linkf href="/">Link 2</Linkf>
                        {/* <Linkf href="/">Link 3</Linkf> */}
                        {/* <Linkf href="/">Link 4</Linkf> */}
                    </Links>
                    <Products>
                        <h1>Name of Product</h1>
                        <Linkf href="/">Product 1</Linkf>
                        <Linkf href="/">Product 2</Linkf>
                        {/* <Linkf href="/">Product 3</Linkf> */}
                        {/* <Linkf href="/">Product 4</Linkf> */}
                    </Products>
                    <Services>
                        <h1>Link of Services</h1>
                        <Linkf href="/">Link 1</Linkf>
                        <Linkf href="/">Link 2</Linkf>
                        {/* <Linkf href="/">Link 3</Linkf> */}
                        {/* <Linkf href="/">Link 4</Linkf> */}
                    </Services>
                </Content>
            </DivBox>
        </Div>
    )
}

export default Footer
```

Footer Part of Homepage

Api's used in project

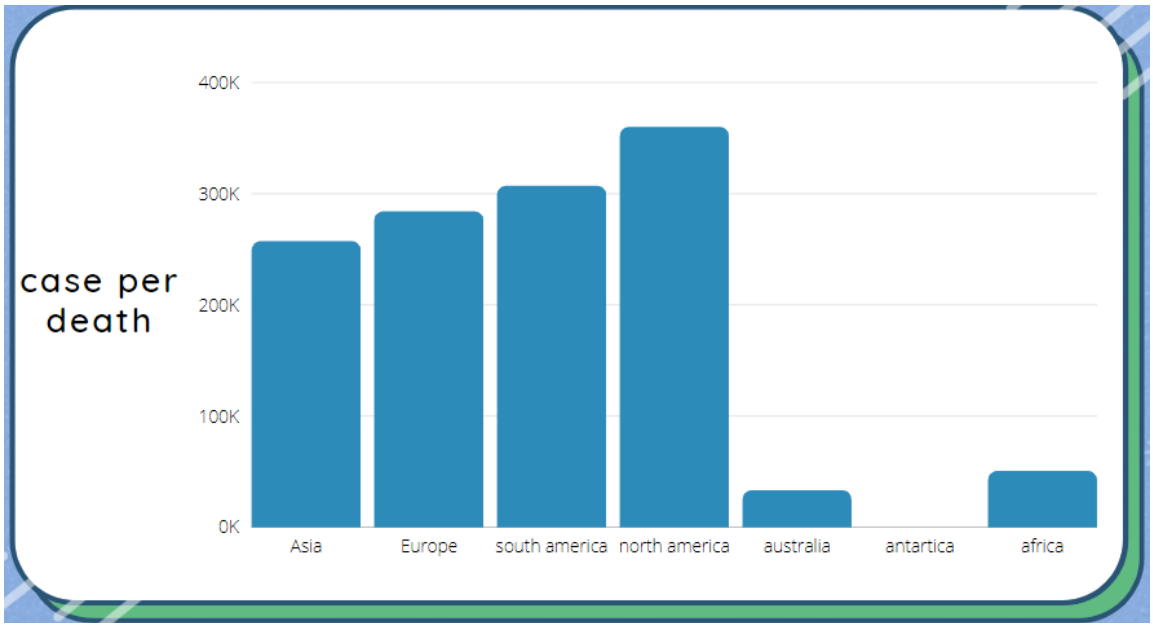
Our Motive to build this Application

- To study COVID-19 data to avoid believing misconceptions that the media and those in power try to impose.
- Learn how to interpret different visualizations
- Assess to see how close it is to possibly predict cases and deaths How?
- Use visualization tools and time series prediction algorithms

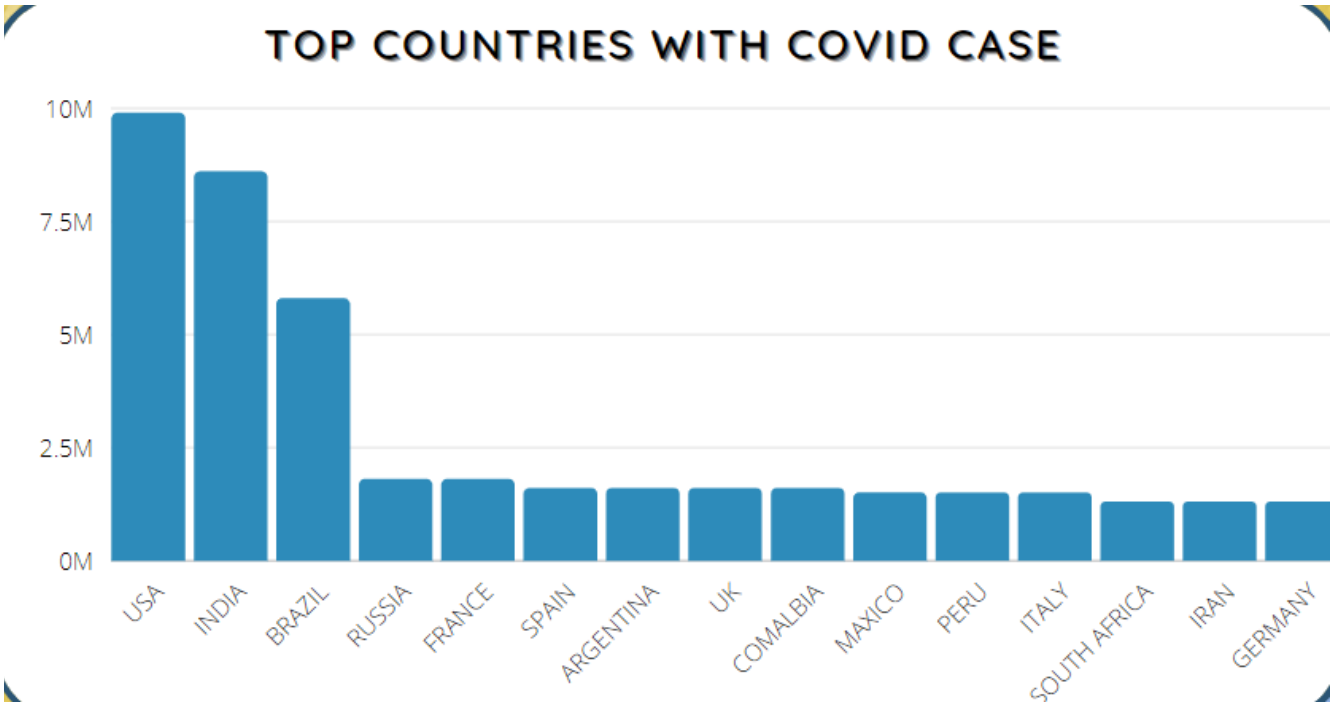
Why Should We Care About Covid19

- A lot of people lost their lives.
- In New Delhi alone, as of December 9, there has been 17,608 deaths attribute
- to the disease.
- The economic condition of the country gets very weak due to covid -19.
- A lot of people lost their job.
- The unemployment rate peaked at an unprecedented level, not seen since data collection started in 1948, in April 2020 (14.7%) before declining to a still-elevated level in November (6.7%).
- In April, every state and the District of INDIA reached unemployment rates greater than their highest unemployment rates during the Great Recession.

Visualizing Data world wide



Visualizing Data by Countries



Testing

nextjs uses many testing libraries from that cypress, jest, and other react testing library.
Cypress is a test runner used for End-to-End (E2E) and Integration Testing.

You can use create-next-app with the with-cypress example to quickly get started.

npx create-next-app@latest --example with-cypress with-cypress-app

ince Cypress is testing a real Next.js application, it requires the Next.js server to be running prior to starting Cypress. We recommend running your tests against your production code to more closely resemble how your application will behave.

Run npm run build and npm run start, then run npm run cypress in another terminal window to start Cypress.

Cypress consists of a free, open source, locally installed Test Runner and a Dashboard Service for recording your tests.

First: Cypress helps you set up and start writing tests every day while you build your application locally. TDD at its best!

Later: After building up a suite of tests and integrating Cypress with your CI Provider, our Dashboard Service can record your test runs. You'll never have to wonder: Why did this fail?

Cypress comes fully baked, batteries included. Here is a list of things it can do that no other testing framework can:

Time Travel: Cypress takes snapshots as your tests run. Hover over commands in the Command Log to see exactly what happened at each step.

Debuggability: Stop guessing why your tests are failing. Debug directly from familiar tools like Developer Tools. Our readable errors and stack traces make debugging lightning fast.
Automatic Waiting: Never add waits or sleeps to your tests. Cypress automatically waits for commands and assertions before moving on. No more async hell.

Spies, Stubs, and Clocks: Verify and control the behaviour of functions, server responses, or timers. The same functionality you love from unit testing is right at your fingertips.
Network Traffic Control: Easily control, stub, and test edge cases without involving your server. You can stub network traffic however you like.

Consistent Results: Our architecture doesn't use Selenium or WebDriver. Say hello to fast, consistent and reliable tests that are flake-free.

Cypress is most often compared to Selenium; however Cypress is both fundamentally and architecturally different. Cypress is not constrained by the same restrictions as Selenium.

This enables you to write faster, easier and more reliable tests.

Screenshots and Videos: View screenshots were taken automatically on failure, or videos of your entire test suite when run from the CLI.
Cross-browser Testing: Run tests within Firefox and Chrome-family browsers (including Edge and Electron) locally and optimally in a Continuous Integration pipeline.

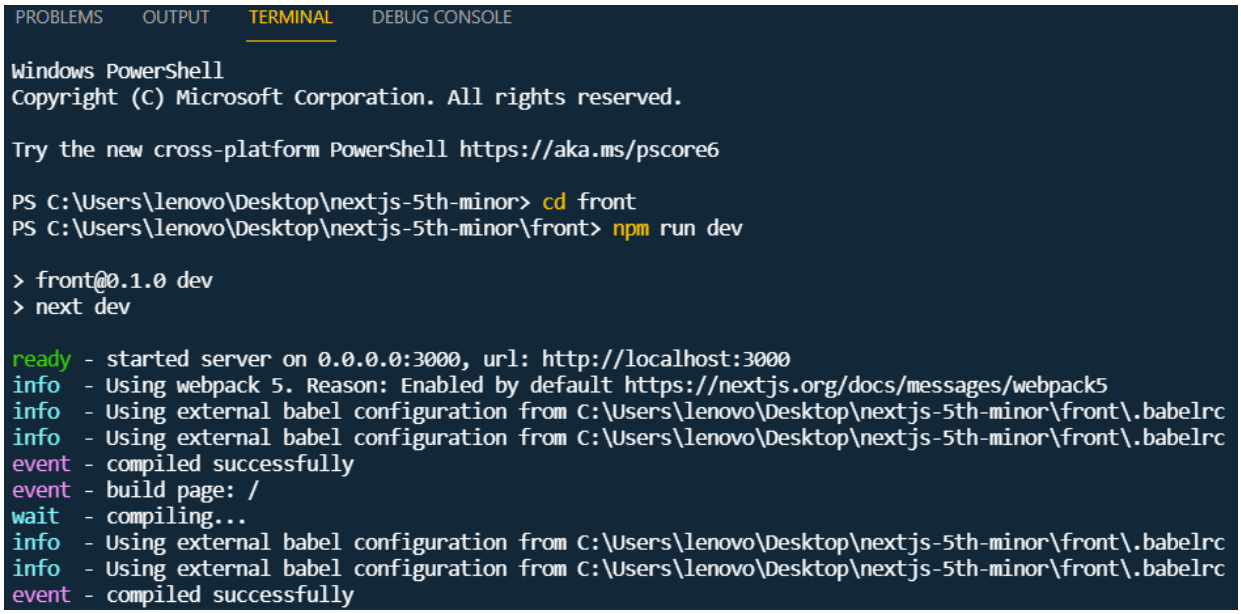
Another framework is available for testing is Playwright.

Playwright is a testing framework that lets you automate Chromium, Firefox, and WebKit with a single API. You can use it to write End-to-End (E2E) and Integration tests across all platforms.

You can get that by run the following command through npx

```
npx create-next-app@latest --example with-playwright with-playwright-app
```

After creating "node modules"
from the command: npm install
(if you are using yarn then run: yarn)



Conclusion & Limitations

The coronavirus disease continues to spread across the world following a trajectory that is difficult to predict. The health, humanitarian and socio-economic policies adopted by countries
will determine the speed and strength of the recovery.
The Covid-Visualizer framework presented in this brief provides a attractive user interface with not only for countries as they faces
the various phases of the covid19 crisis, but also for the international community as a whole. There must be a global human-centred response which is grounded in solidarity.

Coronavirus (COVID-19) Awareness

In light of the Coronavirus (COVID-19) outbreak, we recommend checking the World Health Organization's website for the most accurate and up-to-date information.

We've also created a series of templates to help you share important health information, faster.

Stay safe and healthy.

Reference/Bibliography

There is many reference website for next.js .but. the following official documentation website is better to start.

https://nextjs.org/blog/new-documentation