**ACKNOWLEDGEMENT**

The knowledge and satisfaction that accompany the successful completion of my task would be incomplete without the mention of people who made it possible, whose guidance and encouragement crowned my effort of success. I would like to thank all and acknowledge the help I have received to carry out this mini project.

**Mohammed isfahan shariff**

**1BI19et026**

**Contents**

Page No.

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** |  | **Introduction** | 01 |
| 1.1 | General overview | 01 |
| 1.2 | Problem Statement | 01 |
| **2.** |  | **Back end Design** | 02-05 |
| 2.1 | Conceptual Database Design | 02 |
| 2.2 | Logical Database Design | 03 |
| 2.3 | Normalization | 04-05 |
| **3.** |  | **Front End Design** | 06-07 |
| 3.1 | Screen layout Design | 06 |
| 3.2 | Connectivity | 07 |
| **4.** |  | **Modules** | 08 |
| 4.1 | Models | 08 |
| **5.** |  | **Implementation** | 09-30 |
| **6.** |  | **Snapshots** | 31-34 |
| **7.** |  | **Applications** | 35 |
| **8.** |  | **Conclusion** | 36 |

**CHAPTER-1**

**INTRODUCTION**

**INTRODUCTION**

**1.1 General Overview**

I have chosen to make a Covid-19 Dashboard. In this project, users can see the covid-19 live updates of the last seven days of India. In this project a user can filter state wise data and also can see various graphs related to them .

**1.2 Problem Statement**

Creating a web interface for Users to see live covid updates of the country . To create a web interface which stores the data of the last seven days and displays it in an organised manner . Displaying the state wise covid19 data with various types of graphs to visualize the data



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 1 |

**CHAPTER-2**

**BACK END DESIGN**

**BACK-END DESIGN**

**2.1 ER Diagram**

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

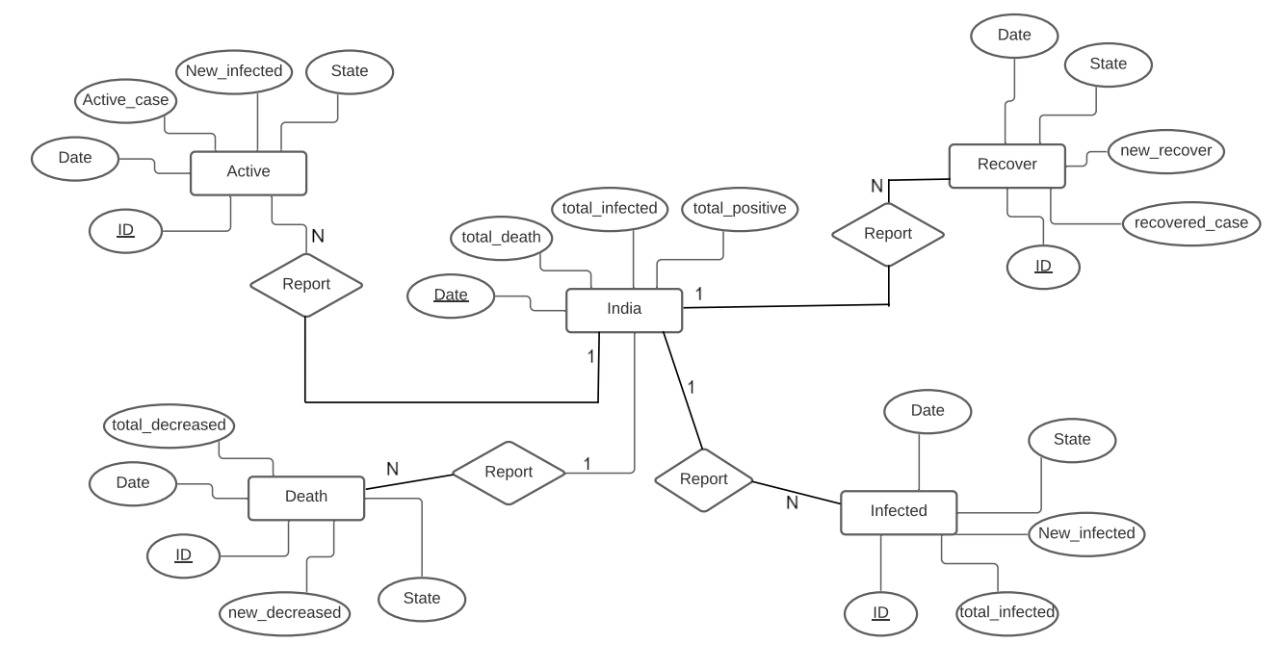


Figure 2.1: ER Diagram for covid database management



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 2 |

Covid-19 Dashboard 1BI19CS147



**2.2 Logical database design**

**Er mapping:**

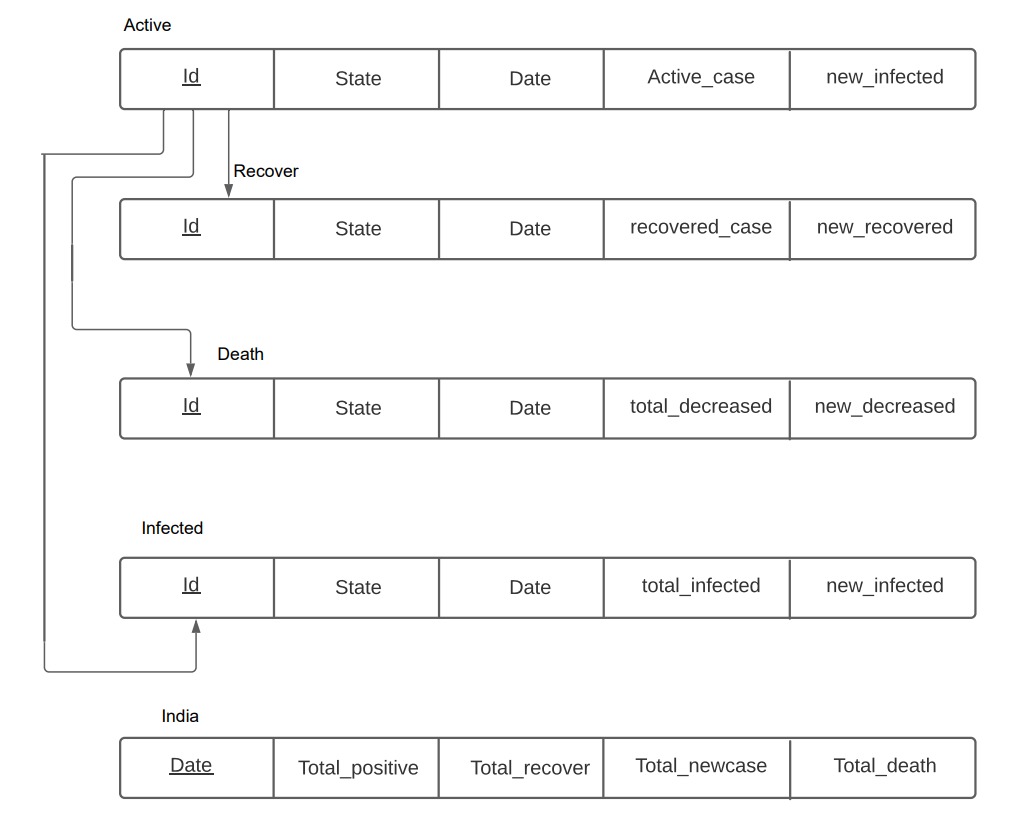


Figure 2.1: Er mapping of covid databa



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 3 |

Covid-19 Dashboard 1BI19CS147



**2.3 Normalisation**

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them relationships

**India**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| date | total\_death | total\_newcase | total\_recovery | total\_positive |
| 2022-01-26 | 491127 | 2223018 | 37370971 | 40085116 |

1NF: The table satisfies 1NF as all attributes have atomic values.

2NF: The table satisfies 2NF as there are no partial dependencies.

3NF: The table satisfies 3 NF as there are no transitive dependencies.

**Active**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | state | date | active\_case | new\_infected |
| 829 | Andaman and Nicobar Islands | 2022-01-26 | 574 | -24 |

1NF: The table satisfies 1NF as all attributes have atomic values.

2NF: The table satisfies 2NF as there are no partial dependencies.

3NF: The table satisfies 3 NF as there are no transitive dependencies.

**Infected**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | state | date | total\_infected | new\_infected |
| 830 | Andhra Pradesh | 2022-01-26 | 2208955 | 70 |

1NF: The table satisfies 1NF as all attributes have atomic values.

2NF: The table satisfies 2NF as there are no partial dependencies.

3NF: The table satisfies 3 NF as there are no transitive dependencies.



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 4 |

Covid-19 Dashboard 1BI19CS147



**Recover**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | state | date | recovered\_case | new\_recovered |
| 831 | Arunachal Pradesh | 2022-01-26 | 56851 | 375 |

1NF: The table satisfies 1NF as all attributes have atomic values.

2NF: The table satisfies 2NF as there are no partial dependencies.

3NF: The table satisfies 3 NF as there are no transitive dependencies.

**Death**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | state | date | decreased | new\_decreased |
| 832 | Assam | 2022-01-26 | 6338 | 19 |

1NF: The table satisfies 1NF as all attributes have atomic values.

2NF: The table satisfies 2NF as there are no partial dependencies.

3NF: The table satisfies 3 NF as there are no transitive dependencies.



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 5 |

**CHAPTER-3**

**FRONT END DESIGN**

**FRONT END DESIGN**

**HTML (HyperText Mark-up Language)**

HTML is a standard mark-up language for creating web pages and web applications with Cascading Style Sheet (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

**CSS (Cascading Style Sheet)**

CSS is a style sheet language used for describing the presentation of a document

written in a mark-up language like HTML. CSS is a cornerstone technology of the World

Wide Web, alongside HTML and JavaScript.

**Django**

Django is Python-based free and open-source web framework, which follows the model-template-view (MVT) architectural pattern. In Django, Python is used throughout, even for settings files and data models.

**3.1 Screen Layout Design**

**HTML <form> TAG**

The HTML <form> element represents a document section that contains interactive controls to submit information to a web server. It is possible to use the :valid and :invalid CSS pseudo-classes to style a <form>element. The HTTP method that the browser uses to submit the form.

**POST**

Corresponds to the HTTP POST method; form data are included in the body of the form and sent to the server.

**GET**

Corresponds to the HTTP GET method; form data is appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server. This method is used when the form has no side-effects and contains only ASCII characters. This value can be overridden by a form method attribute on a <button> or <input> element.



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 6 |

Covid-19 Dashboard 1BI19CS147



**Action**

The URI of a program that processes the form information. This value can be overridden by a form action attribute on a <button> or <input> element.

**HTML <input> TAG**

The HTML <input> element is used to create interactive controls for web-based forms in order to accept data from the user. An <input> work varies considerably depending on the value of its type attribute; hence the different types are covered in their own separate reference pages. If this attribute is not specified, the default type adopted type is text.

**3.2 Connectivity to MySQL database**

We need our MySQL server address (if the database is on the same server as the web server it will most likely be localhost or 127.0.0.1), username, password and database name. The connectivity is done using mysql-connector-python. The connection is done each time when the admin wants to retrieve anything from the database, or add something to the database. All in all, whenever the admin wants to perform an operation wherein the database is involved, the connection is required. The below code must be written inside the file Settings.py.   
 DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'dbms\_project',

'HOST': '127.0.0.1',

'PORT': '3306',

'USER': 'root',

'PASSWORD': '123456789',

}

}



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 7 |

Covid-19 Dashboard 1BI19CS147





|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 9 |

**CHAPTER-4**

**MAJOR MODULES**

**MAJOR MODULES**

1. **Index.html :**

This is the main page of the project . Where a user can see the last updated data of Covid19 of India . The page has various data like total active cases, total death ,total infected and total recovered cases . Not only the total data can be seen but also the change since 24 hrs can be seen . The data is taken from the Ministry of Health and Family welfare site . In this page you can also see the state wise covid19 update .

**2. India.html / State.html:**

In this page a user can see the covid update of a particular state by typing the state name in the search bar displayed in the Index.html page. After the user searches for a state which is a valid name , then they will be redirected to this page in which they can see the record of past seven days with various user friendly graphs and a tabular arrangement of covid data of the state .

If the user clicks on covid India on the index page then they will be redirected to India.html page where they can see the detailed information of past seven days with graphs .

**3. Awareness.html:**

In this page various awareness is mentioned related to covid 19. Things to do to control the spread of the virus . How to protect yourself from the virus and all sorts of safety methods are mentioned .



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 8 |

**CHAPTER-5**

**IMPLEMENTATION**

**IMPLEMENTATION**

**5.1 Front-end Implementation**

HTML is a standard mark-up language for creating web pages and web applications with Cascading Style Sheet (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

**5.1.1 Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

{% load static %}

{% load mathfilters %}

<link rel="stylesheet" type="text/css" href="{% static 'des.css'%}">

<scriptsrc="https://kit.fotawesome.com/87f488c.js"crossorigin="anonymous"></script>

<style> .navbar .search-container{

float: right; padding-top: 20px; }

.navbar .search-container button {

float: right; padding: 6px 10px;

margin-right: 16px; background: #ddd; font-size: 17px; border: none;

cursor: pointer; }

.navbar .search-container button:hover { background: #ccc;}

.navbar input[type=text] {padding: 6px; font-size: 17px; border: none; }

</style><title></title></head><body class = 'body'> <div class="navbar">

<a href="#body">Home</a>

<a href="{% url 'help' %}" >Awareness</a>

<a style="margin-left: 450px;" href="{% url 'Ind' %}"> Covid India</a>

<div class="search-container">

<form action="{% url 'search-state' %}" method="post" >

{% csrf\_token %}



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 9 |

Covid-19 Dashboard 1BI19CS147



<input type="text" placeholder="State name.." name="search">

<button type="submit"><i class="fa fa-search"></i></button>

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

<div class="row" id="body">

<div class="Positive" style="font-size: 25px;" >Positive Case<br>

<p style="float: left; padding-left: 15px; padding-top: 8px;">New Update =</p>

<p style="padding-right: 50px; padding-bottom: 8px; padding-top: 8px;

color: red;"> {{india\_data.0.new\_infected\_\_sum }}&#8593<br></p>

<p style="float: left; padding-left: 15px;">Total =</p>

<p style="padding-right: 105px;">{{total.0.4}} </p></div>

<div class="Death" style="font-size: 25px">Active Case<br>

<p style="float: left; padding-left: 15px; padding-top: 8px;">New Update =</p>

{% if india\_data.1.new\_infected\_\_sum > 0 %}

<p style="padding-right: 50px; padding-bottom: 8px; padding-top: 8px; color: red;">{{india\_data.1.new\_infected\_\_sum }}&#8593<br></p>

{% else %}

<p style="padding-right: 50px; padding-bottom: 8px; padding-top: 8px;color: green;">{{india\_data.1.new\_infected\_\_sum|abs }}&#8595<br> </p>

{% endif %}

<p style="float: left; padding-left: 15px;">Total =</p>

<p style="padding-right: 105px;">

{{total.0.3}}</p> </div>

<div class="Recovered" style="font-size: 25px">Death<br>

<p style="float: left; padding-left: 35px; padding-top: 8px;">New Update =</p>

<p style="padding-right: 50px; padding-bottom: 8px; padding-top: 8px; color: red;">{{india\_data.3.new\_decreased\_\_sum}}&#8593<br></p>

<p style="float: left; padding-left: 35px;">Total =</p>

<p style="padding-right: 105px;">{{total.0.1}}</p></div>

<div class="Recovered" style="font-size: 25px">Recovery<br>

<p style="float: left; padding-left: 25px; padding-top: 8px;">New Update =</p>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 10 |

Covid-19 Dashboard 1BI19CS147



<p style="padding-right: 30px; padding-bottom: 8px; padding-top: 8px; color: green;">{{india\_data.2.new\_recovered\_\_sum}}&#8593<br></p>

<p style="float: left; padding-left: 25px;">Total =</p>

<p style="padding-right: 105px;">{{total.0.2}}</p></div></div>

<div class="Update">

<a onclick="location.href='{% url 'get\_data' %}'"><button class="butt">Update</button></a><p>

<span class="blink">Last updated on : {{output\_data}}</span></p>

</div><div class="State" style="width: 95%;">

<p style="font-size: 30px; font-weight: bold; padding: 15px 32px; padding-left: 50px;">COVID-19 Statewise Status<p>

<table class="covid" style="margin-right: 15px; margin-left: 50px; " >

<thead> <tr>

<th style="width: 4%;">S.No.</th>

<th style="width: 20%;">Name of State/UT</th>

<th style="width: 19%;">Infected Case

<table class="covid">

<th style="width: 50%;">Total</th>

<th style="width: 50%;">Change since yesterday</th>

</table></th>

<th style="width: 19%;">Active Cases

<table class="covid">

<th style="width: 50%;">Total</th>

<th style="width: 50%;">Change since yesterday</th>

</table></th>

<th style="width: 19%;">Cured/Discharged

<table class="covid">

<th style="width: 50%;">Total</th>

<th style="width: 50%;">Change since yesterday</th>

</table>

</th>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 11 |

Covid-19 Dashboard 1BI19CS147



<th style="width: 19%;">Death

<table class="covid">

<th style="width: 50%;">Total</th>

<th style="width: 50%;">Change since yesterday</th>

</table></th> </tr></thead>

{% for a, b in data %}

<tr><td>{{forloop.counter}}</td><td>{{a.0}}</td> <td>

<table class ='covid\_data' ><tr>

<th style="width: 50%;">{{a.4}}</th>

{% if a.5 >= 0 %}

<th style="width: 50%; color:red;">{{a.5}} &#8593;</th>

{% else %}

<th style="width: 50%; color:limegreen;">

{{a.5|abs}} &#8593;</th>

{% endif %}</tr> </table></td><td>

<table class ='covid\_data' ><tr>

<th style="width: 50%;">{{a.2}}</th>

{% if a.3 >= 0 %}

<th style="width: 50%; color: red; ">{{a.3}} &#8593</th>

{% else %}

<th style="width: 50%; color:limegreen; ">

{{ a.3|abs }} &#8595;</th>

{% endif %}</tr></table></td><td>

<table class ='covid\_data' ><tr>

<th style="width: 50%;">{{b.2}}</th>

{% if b.3 >= 0 %}

<th style="width: 50%; color:limegreen">{{b.3}} &#8593</th>

{% else %}

<th style="width: 50%; color:red">{{b.3}} &#8593</th>

{% endif %}</tr></table></td><td>

<table class ='covid\_data' ><tr>

<th style="width: 50%;">{{b.4}}</th>

|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 12 |

Covid-19 Dashboard 1BI19CS147



{% if b.5 > 0 %}

<th style="width: 50%; color: red;">{{b.5}} &#8593</th>

{% else %}

<th style="width: 50%; ">{{b.5}}</th>

{% endif %}</tr></table></td> </tr>{% endfor %}</table></div>

<footer><p>Disclaimer:

<p>Website content is taken from:</p>

<a href="https://www.mohfw.gov.in/" style="color: darkorange;"> Ministry of Health and Family Welfare</p>

<p style="color: aliceblue;">Design and Developed By:Shubham & Suffiyan</p> </p></footer></body></html>

**5.1.2 Help.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

{% load static %}

<link rel="stylesheet" type="text/css" href="{% static 'hh.css'%}">

<title>Prevention of Coronavirus</title>

<title></title> </head>

<body> <h1>How to Protect Yourself</h1>

<!-- Corona animation container -->

<div class="animatediv">

<img src="{% static 'pictures/corona-virus.png' %}"></div>

<!-- Image on top of the Main menu container -->

<img src="{% static 'pictures/people-mask.png' %}" class="TopImage">

<!-- Main Menu Start -->

<ul class="menu shadow"><li>

<a href="#CleanYourHand">Clean your hands often</a> </li><li>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 13 |

Covid-19 Dashboard 1BI19CS147



<a href="#AvoidCloseContact">Avoid close contact</a></li> <li>

<a href="#StayHome">Stay home if you’re sick</a></li><li>

<a href="#CoverCoughs">Cover coughs and sneezes</a></li><li>

<a href="#WearAFacemask">Wear a facemask if you are sick</a> </li><li>

<a href="#CleanAndDisinfect">Clean and Disinfect </a></li></ul>

<!-- Free Space container -->

<div style="margin-top:40%"></div>

<!-- Message Boxes -->

<h2>Clean Your Hand</h2>

<div class="messagebox shadow" id="CleanYourHand"><ul>

<li><strong>Wash your hands</strong> often with soap and water for at least 20 seconds especially after you have been in a public place, or after blowing your nose, coughing, or sneezing.</li><br>

<li>If soap and water are not readily available, <strong>use a hand sanitizer that contains at least 60% alcohol</strong>. Cover all surfaces of your hands and rub them together until they feel dry.</li> <br>

<li><strong>Avoid touching</strong> <strong>your eyes, nose, and mouth</strong> with unwashed hands.</li> </ul></div> </div>

<h2>Avoid Close Contact</h2>

<div class="messagebox shadow" id="AvoidCloseContact"><ul>

<li><strong>Avoid close contact</strong> with people who are sick</li>

<li>Put <strong>distance between yourself and other</strong> <strong>people</strong> if COVID-19 is spreading in your community. This is especially important for people who are at higher risk of getting very sick.</li></ul></div><h2>Stay Home</h2>

<div class="messagebox shadow" id="StayHome">

<li><strong>Stay home</strong> if you are sick, except to get medical care. Learn what to do if you are sick.</li> </div><h2>Cover Coughs</h2>

<div class="messagebox shadow" id="CoverCoughs"><ul>

<li><strong>Cover your mouth and nose</strong> with a tissue when you cough or sneeze or use the inside of your elbow.</li>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 14 |

Covid-19 Dashboard 1BI19CS147



<li><strong>Throw used tissues</strong> in the trash.</li>

<li>Immediately <strong>wash your hands</strong> with soap and water for at least 20 seconds. If soap and water are not readily available, clean your hands with a hand sanitizer that contains at least 60% alcohol.</li></ul></div>

<h2>Wear A Facemask</h2>

<div class="messagebox shadow" id="WearAFacemask"><ul>

<li><strong>If you are sick:</strong> You should wear a facemask when you are around other people (e.g., sharing a room or vehicle) and before you enter a healthcare provider’s office. If you are not able to wear a facemask (for example, because it causes trouble breathing), then you should do your best to cover your coughs and sneezes, and people who are caring for you should wear a facemask if they enter your room.</li>

<li><strong>If you are NOT sick: </strong>You do not need to wear a facemask unless you are caring for someone who is sick (and they are not able to wear a facemask). Facemasks may be in short supply and they should be saved for caregivers.</li></ul> </div> <h2>Clean And Disinfect</h2>

<div class="messagebox shadow" id="CleanAndDisinfect"><ul>

<li><strong>Clean AND disinfect frequently touched surfaces daily</strong>. This includes tables, doorknobs, light switches, countertops, handles, desks, phones, keyboards, toilets, faucets, and sinks.</li>

<li><strong>If surfaces are dirty, clean them:</strong> Use detergent or soap and water prior to disinfection.</li></ul></div>

<div id="WashHandsTxt">Wash Your Hands</div><!-- Table Container -->

<table class="Tbl"><tr><th>WHO</th> <th>CORONAVIRUS OUTBREAK</th>

<th>MINISTRY OF HEALTH INDIA</th></tr><tr><td>

<ahref="<https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance>">

<img src="{% static 'pictures/uno.png' %}" alt="world health organization" style="width:100px"></a></td><td>

<a href="https://www.worldometers.info/coronavirus/">

<img src="{% static 'pictures/world.gif' %}"> </a></td><td>

<a href="https://www.mohfw.gov.in/#">

<img src="{% static 'pictures/india.png' %}" style="width:60px"</a> </td>

</tr> </table> </div> </body>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 15 |

Covid-19 Dashboard 1BI19CS147



**5.1.3 State.html / India.html**

<!DOCTYPE html><html lang="en"><head>{% load static %}{% load mathfilters %}

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<link rel="stylesheet" type="text/css" href="{% static 'sta.css'%}">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/chart.js@2.9.4"></script>

<title>{{search}}</title></head><body>

<div class="State" style="width: 95%;">

<p style="font-size: 30px; font-weight: bold; padding: 15px 32px; padding-left: 50px;">Last week COVID-19 Status of {{search}}<p>

<table class="covid" style="margin-right: 15px; margin-left: 50px; " ><thead>

<tr> <th style="width: 4%;">S.No.</th> <th style="width: 20%;">Date</th>

<th style="width: 19%;">Infected Case</th>

<th style="width: 19%;">Active Cases </th>

<th style="width: 19%;">Cured/Discharged</th>

<th style="width: 19%;">Death</th> </tr></thead>

{% for a,b,c,d in final %}<tr> <td>{{forloop.counter}}</td>

<td>{{a.date}}</td>{% if d.new\_infected >= 0 %}

<td style="color: red;">{{d.new\_infected}} &#8593</td> {% else %}

<td style="color: green;">{{d.new\_infected}} &#8595</td>{% endif %}

{% if a.new\_infected >= 0 %}

<td style="color:red;">{{a.new\_infected}} &#8593</td>

{% else %} <td style="color:green;">{{a.new\_infected|abs}} &#8595</td>

{% endif %} <td style="color: green;">{{b.new\_recovered}} &#8593</td>

<td style="color:red;">{{c.new\_decreased}} &#8593</td> </tr>

{% endfor %}</table> </div> <div class ="chart"><div>

<canvas id="myChart" width="100" height="30"></canvas></div><script>

const labels = [{% for x in xval %} '{{x}}', {% endfor %}]; const data = {

labels: labels,datasets: [{label: 'Infected',backgroundColor: 'transparent',



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 16 |

Covid-19 Dashboard 1BI19CS147



data: [{% for y in inf %} '{{y}}', {% endfor %}],}, {label: 'Cured',

backgroundColor: 'transparent', borderColor: 'rgb(24, 106, 59)',

data: [{% for y in reco %} '{{y}}',{% endfor %}], }, { label: 'Active',

backgroundColor: 'transparent', borderColor: 'rgb(93, 173, 226)',

data: [{% for y in act %} '{{y}}',{% endfor %}], },{

label: 'Death', backgroundColor: 'transparent',borderColor: 'rgb(186, 74, 0)',

data: [{% for y in dea %} '{{y}}',{% endfor %}],} ]};const config = {

type: 'line', data: data,options: {} };</script><script>

const myChart = new Chart(document.getElementById('myChart'),config);

</script> </div> <div class ="chart"> <canvas id="chart" width="100" height="30"></canvas><script>

let ctx = document.getElementById("chart").getContext("2d");

let chart = new Chart(ctx, { type: "bar", data: {

labels: [{% for x in xval %} '{{x}}', {% endfor %}],datasets: [ {

label: 'Infected', backgroundColor: 'rgb(123, 36, 28)',

borderColor: 'rgb(123, 36, 28)',data: [{% for y in inf %} '{{y}}', {% endfor %}],},

{label: 'Cured',

backgroundColor: 'rgb(24, 106, 59)',borderColor: 'rgb(24, 106, 59)',

data: [{% for y in reco %} '{{y}}',{% endfor %}], },{ label: 'Active',

backgroundColor: 'rgb(93, 173, 226)', borderColor: 'rgb(93, 173, 226)',

data: [{% for y in act %} '{{y}}',{% endfor %}],},{ label: 'Death',

backgroundColor: 'rgb(186, 74, 0)',borderColor: 'rgb(186, 74, 0)',

data: [{% for y in dea %} '{{y}}',{% endfor %}],} ]},options: {

title: { text: "Bar-Chart",display: true} } });</script> </div><div>

<canvas id="pieChart" style="margin-left: 10px; padding-top: 30px;" width="100" height="20" ></canvas> <script>

var barColors = ["#b91d47", "#00aba9","#2b5797", "#e8c3b9",

"#1e7145","#D4AC0D", "#8E44AD"];

new Chart("pieChart", {type: "pie",data: {

labels: [{% for x in xval %} '{{x}}', {% endfor %}],



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 17 |

Covid-19 Dashboard 1BI19CS147



datasets: [{

backgroundColor: barColors,

data: [{% for y in act %} '{{y}}',{% endfor %}] } ]},

options: {title: {display: true, text: "Active" }} });

</script> </div><div>

<canvas id="pieChart1" style="margin-left: 10px; padding-top: 30px;" width="100" height="20" ></canvas> <script>

var barColors = [ "#b91d47", "#00aba9", "#2b5797", "#e8c3b9","#1e7145",

"#D4AC0D", "#8E44AD" ];

new Chart("pieChart1", { type: "pie",data: {

labels: [{% for x in xval %} '{{x}}', {% endfor %}],

datasets: [{backgroundColor: barColors,

data: [{% for y in inf %} '{{y}}',{% endfor %}]}]},

options: {title: { display: true, text: "Infected" } } }); </script></div><div>

<canvas id="pieChart2" style="margin-left: 10px; padding-top: 30px;" width="100" height="20" ></canvas>

<script> var barColors = [ "#b91d47", "#00aba9", "#2b5797","#e8c3b9",

"#1e7145","#D4AC0D","#8E44AD"];

new Chart("pieChart2", {type: "pie",

data: { labels: [{% for x in xval %} '{{x}}', {% endfor %}],

datasets: [{ backgroundColor: barColors,

data: [{% for y in reco %} '{{y}}',{% endfor %}]} ]},

options: { title: { display: true, text: "Recovery }} });

</script>

</div>

<div>

<canvas id="pieChart3" style="margin-left: 10px; padding-top: 30px;" width="100" height="20" ></canvas><script>

var barColors = ["#b91d47", "#00aba9","#2b5797", "#e8c3b9","#1e7145",

"#D4AC0D","#8E44AD"];

new Chart("pieChart3", { type: "pie",

|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 18 |



Covid-19 Dashboard 1BI19CS147



data: { labels: [{% for x in xval %} '{{x}}', {% endfor %}],datasets: [{

backgroundColor: barColors, data: [{% for y in dea %} '{{y}}',{% endfor %}] }] },

options: {title: {display: true,text: "Death" } }});</script>

</div>

<footer>

<p>Disclaimer:

<p>Website content is taken from:</p>

<a href="https://www.mohfw.gov.in/" style="color: darkorange;"> Ministry of Health and Family Welfare</p>

<p style="color: aliceblue;">Design and Developed By:Shubham & Suffiyan</p>

</p> </footer>

</body>

</html>



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 19 |

Covid-19 Dashboard 1BI19CS147



**5.2.1 Table creation**

a)India

CREATE TABLE India

( date DATE NOT NULL,

state VARCHAR(100) NOT NULL,

total\_death NUMBER(10) NOT NULL,

total\_newcase NUMBER(10) NOT NULL,

total\_recovery NUMBER(10) NOT NULL,

total\_positive NUMBER(10) NOT NULL,

CONSTRAINT In PRIMARY KEY (date)

);

b)Active

CREATE TABLE Active

(id NUMBER(4) NOT NULL,

state VARCHAR(100) NOT NULL,

date DATE NOT NULL,

active\_case NUMBER(10) NOT NULL,

new\_infected NUMBER(10) NOT NULL,

CONSTRAINT Ac PRIMARY KEY (id)

);

c)Infected

CREATE TABLE infected

(id NUMBER(4) NOT NULL,

state VARCHAR(100) NOT NULL,

date DATE NOT NULL,

total\_infected NUMBER(10) NOT NULL,

new\_infected NUMBER(10) NOT NULL,

CONSTRAINT FC FOREIGN KEY (id) REFERENCES active(id) );



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 20 |

Covid-19 Dashboard 1BI19CS147



d)Death

CREATE TABLE death

(id NUMBER(4) NOT NULL,

state VARCHAR(100) NOT NULL,

date DATE NOT NULL,

decreased NUMBER(10) NOT NULL,

new\_decreased NUMBER(10) NOT NULL,

CONSTRAINT FC FOREIGN KEY (id) REFERENCES active(id) );

e)Recovery

CREATE TABLE recovery

(id NUMBER(4) NOT NULL,

state VARCHAR(100) NOT NULL,

date DATE NOT NULL,

total\_recovered NUMBER(10) NOT NULL,

new\_recovered NUMBER(10) NOT NULL,

CONSTRAINT FC FOREIGN KEY (id) REFERENCES active(id) );

**5.2.2 Stored procedure**

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So, if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

**Stored procedure 1: get\_active\_infected**

delimiter //

create procedure get\_active\_infected(in da date)

begin select

a.state,a.date,a.active\_case,(a.new\_infected)as new\_active,b.total\_infected,b.new\_infected

from dbms\_project\_active a

inner join dbms\_project\_infected b

on a.state=b.state and (a.date=da and b.date=da);

end //



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 21 |

Covid-19 Dashboard 1BI19CS147



**Stored procedure 2: get\_recover\_death**

delimiter //

create procedure get\_recover\_death(in da date)

begin

select s.state,s.date,s.recovered\_case,s.new\_recovered,d.decreased,d.new\_decreased

from dbms\_project\_recover s

inner join dbms\_project\_death d

on s.state=d.state and (s.date=da and d.date=da);

end //

**5.2.3 Trigger**

A trigger is a special type of stored procedure that automatically runs when an event

occurs in the database server. DML triggers run when a user tries to modify data through

a data manipulation language (DML) event. DML events are INSERT, UPDATE, or

DELETE statements on a table or view.

**Trigger 1: delete\_data**

delimiter //

create trigger delete\_data

after insert

on dbms\_project\_india for each row

begin

delete from dbms\_project\_active where date = DATE\_SUB(CURDATE(),INTERVAL 7 DAY);

delete from dbms\_project\_death where date = DATE\_SUB(CURDATE(),INTERVAL 7 DAY);

delete from dbms\_project\_infected where date = DATE\_SUB(CURDATE(),INTERVAL 7 DAY);



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 22 |

Covid-19 Dashboard 1BI19CS147



delete from dbms\_project\_recover where date = DATE\_SUB(CURDATE(),INTERVAL 7 DAY);

end //

**Trigger 2: delete\_data\_india**

delimiter //

create trigger delete\_data\_india

after insert

on dbms\_project\_active for each row

begin

delete from dbms\_project\_india where date = DATE\_SUB(CURDATE(),INTERVAL 7 DAY);

end //



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 23 |

Covid-19 Dashboard 1BI19CS147



**5.2.4 Views.py**

from audioop import reverse

from http.client import HTTPResponse

from math import inf

import re

from socket import \*

from sqlite3 import Cursor

from subprocess import call

from turtle import pos

from django.db import connection

from django.db.models.functions import Cast

from django.db.models.fields import DateField

from django.shortcuts import render

from django.template import loader

from django.http import HttpResponse , HttpResponseRedirect

from django.db.models import Sum

from dbms\_project.models import \*

import json

import requests

from datetime import date

import datetime

today = date.today()

def index(request):

cursor = connection.cursor()

cursor2 = connection.cursor()



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 24 |

Covid-19 Dashboard 1BI19CS147



modify\_=modify.objects.filter(id=666).values('date')

modify\_ =modify\_[0]['date']

cursor.callproc('get\_active\_infected',[modify\_])

result = cursor.fetchall()

result=list(result)

cursor.close()

cursor2.callproc('get\_recover\_death',[modify\_])

result2 = cursor2.fetchall()

result2=list(result2)

cursor2.close()

new\_result = zip(result,result2)

india\_data=find\_sum()

total=india.objects.filter(date=modify\_).values\_list()

return render(request,'index.html',{'data':new\_result,'india\_data':india\_data,"output\_data":modify\_,'total':total})

def run\_data(request):

count\_data ='hello'

date\_=[]

try:

date\_ = india.objects.get(date=today)

count\_data='data updated'

except:

count\_data = 'data stored'



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 25 |

Covid-19 Dashboard 1BI19CS147



URL = "https://www.mohfw.gov.in/data/datanew.json"

page = requests.get(URL)

data=json.loads(page.content)

state\_wise = []

for da in data:

dic ={}

dic['id']=da['sno']

dic['name']=da['state\_name']

dic['new\_active'] = int(da['new\_active'])-int(da['active'])

dic['new\_cured']=int(da['new\_cured'])-int(da['cured'])

if(da['total'] == ''):

dic['new\_death'] = 0

else:

dic['new\_death']=int(da['total'])

dic['new\_positive']=int(da['new\_positive'])-int(da['positive'])

dic['total\_active'] = int(da['new\_active'])

dic['total\_cured'] = int(da['new\_cured'])

dic['total\_death']=int(da['new\_death'])

dic['total\_positive']=int(da['new\_positive'])

state\_wise.append(dic)

for da in state\_wise:

if(da['name'] is not ''):



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 26 |

Covid-19 Dashboard 1BI19CS147



add\_active=active(state=da['name'],date=today,active\_case=da['total\_active'],new\_infected=da['new\_active'])

add\_active.save()

add\_death=death(state=da['name'],date=today,decreased=da['total\_death'],new\_decreased=da['new\_death'])

add\_death.save()

add\_recovery=recover(state=da['name'],date=today,recovered\_case=da['total\_cured'],new\_recovered=da['new\_cured'])

add\_recovery.save()

add\_infected=infected(state=da['name'],date=today,total\_infected=da['total\_positive'],new\_infected=da['new\_positive'])

add\_infected.save()

add\_india=india(date=today,total\_death=state\_wise[-1]['total\_death'],total\_recovery=state\_wise[-1]['total\_cured'],total\_newcase=state\_wise[-1]['total\_active'],total\_positive=state\_wise[-1]['total\_positive'])

add\_india.save()

a =modify.objects.get(id=666)

a.date=today

a.save()

return HttpResponseRedirect('/')



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 27 |

Covid-19 Dashboard 1BI19CS147



def find\_sum():

total=[]

modify\_=modify.objects.filter(id=666).values('date')

modify\_ =modify\_[0]['date']

total.append(infected.objects.filter(date=modify\_).aggregate(Sum('new\_infected')))

total.append(active.objects.filter(date=modify\_).aggregate(Sum('new\_infected')))

total.append(recover.objects.filter(date=modify\_).aggregate(Sum('new\_recovered')))

total.append(death.objects.filter(date=modify\_).aggregate(Sum('new\_decreased')))

return total

def search(request):

search= request.POST['search']

search = search.title()

a = active.objects.filter(state=search).values()

b= recover.objects.filter(state=search).values()

c= death.objects.filter(state=search).values()

d=infected.objects.filter(state=search).values()

final = zip(a,b,c,d)

final1=zip(a,b,c,d)

if(len(d) != 0):

xval=[]

inf = []

dea=[]

reco=[]

act=[]



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 28 |

Covid-19 Dashboard 1BI19CS147



for a,b,c,d in final1:

xval.append(a['date'])

inf.append(d['new\_infected'])

dea.append(c['new\_decreased'])

act.append(abs(a['new\_infected']))

reco.append(b['new\_recovered'])

return render(request,'state.html',{'search':search , 'final':final,'xval':xval,'inf':inf,'dea':dea,'reco':reco,'act':act})

else:

return HttpResponseRedirect('/')

def help(request):

return render(request,'help.html')

def Ind(request):

modify\_=modify.objects.filter(id=666).values('date')

modify\_ =modify\_[0]['date']

data=[]

date=[]

for i in range(6,-1,-1):

data\_temp =[]

new = datetime.timedelta(days=i)

da = modify\_-new

date.append(da)

data\_temp.append(da)

data\_temp.append(infected.objects.filter(date=da).aggregate(Sum('new\_infected')))

data\_temp.append(active.objects.filter(date=da).aggregate(Sum('new\_infected')))



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 29 |

Covid-19 Dashboard 1BI19CS147



data\_temp.append(recover.objects.filter(date=da).aggregate(Sum('new\_recovered')))

data\_temp.append(death.objects.filter(date=da).aggregate(Sum('new\_decreased')))

data.append(data\_temp)

return render(request,'Ind.html',{'data\_':data,'date':date})



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 30 |

**CHAPTER-6**

**SNAPSHOTS**

**SNAPSHOTS**

**6.1 Home page**

This is the main page of the website. It displays the total covid data of the country and also the states . It also shows the change in data or the new case reported in the last 24 hrs

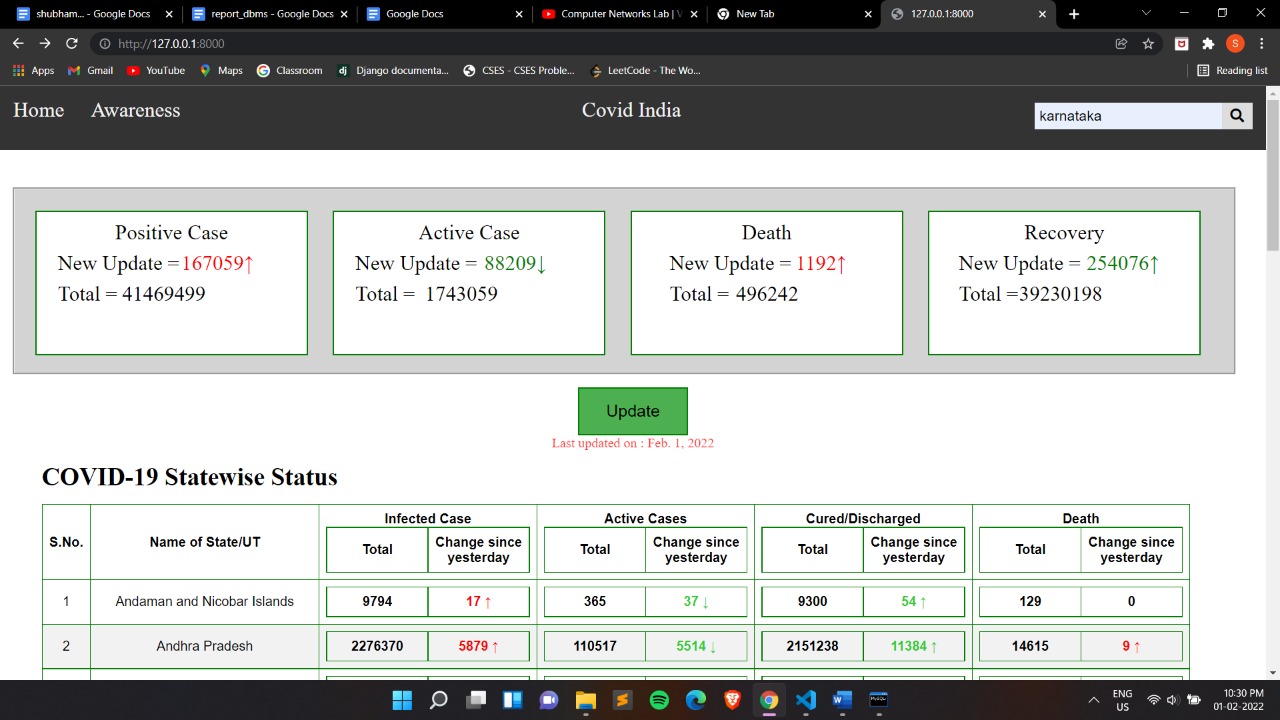


Figure 6.1: Main page

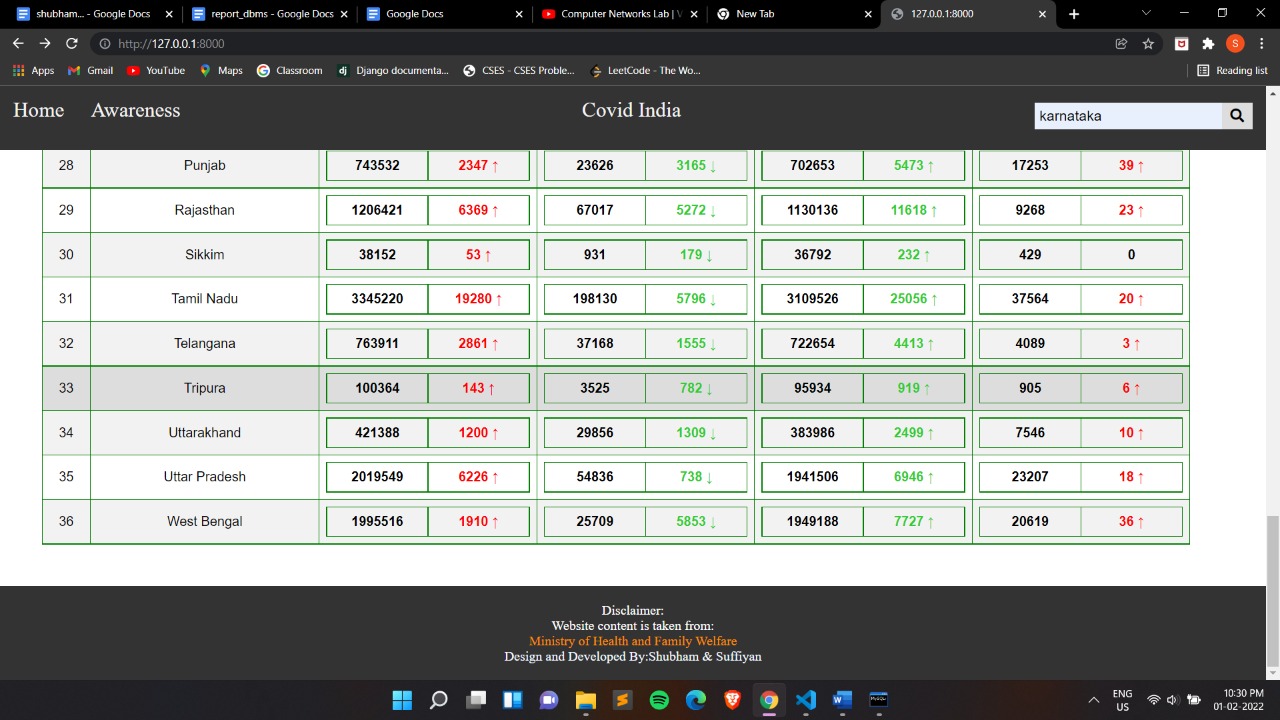


Figure 6.2:State data



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 31 |

Covid-19 Dashboard 1BI19CS147



**6.2 Status of India/State**

This page displays the last 7 days record of covid19 of specified state or India , various types of graphs can also be seen in this page which is related to covid19 data.

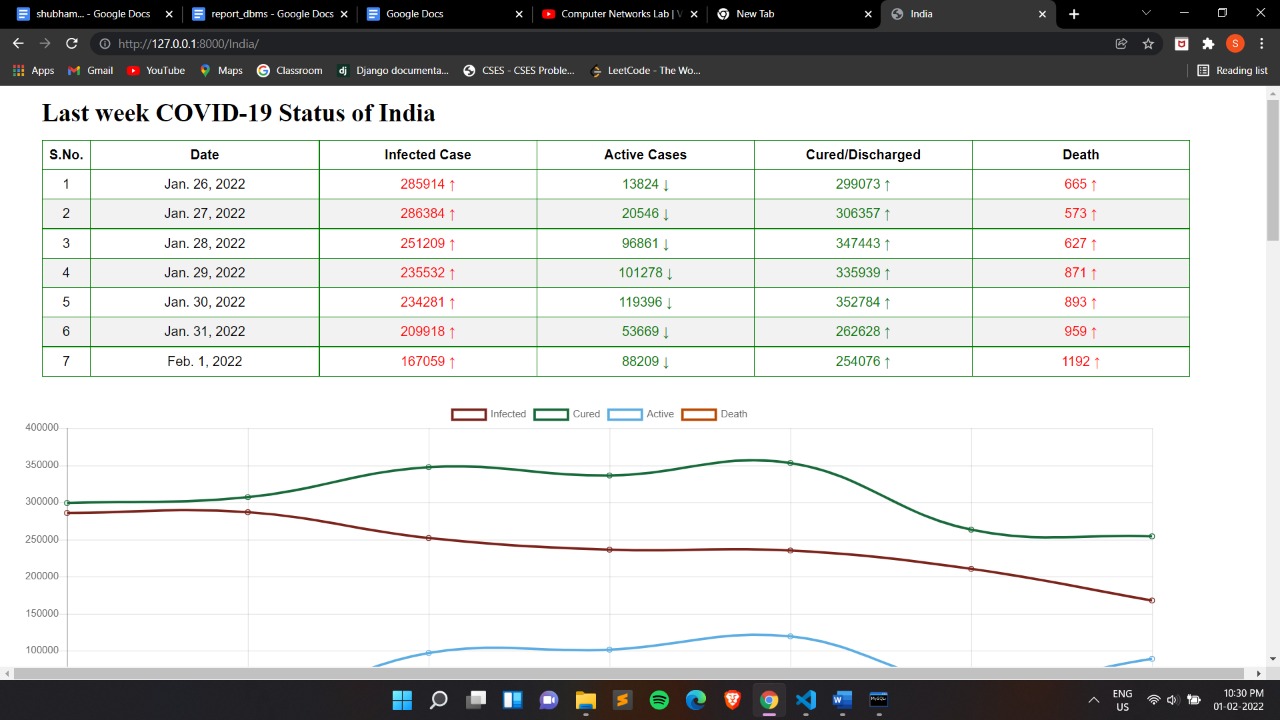


Figure 6.3: State graph

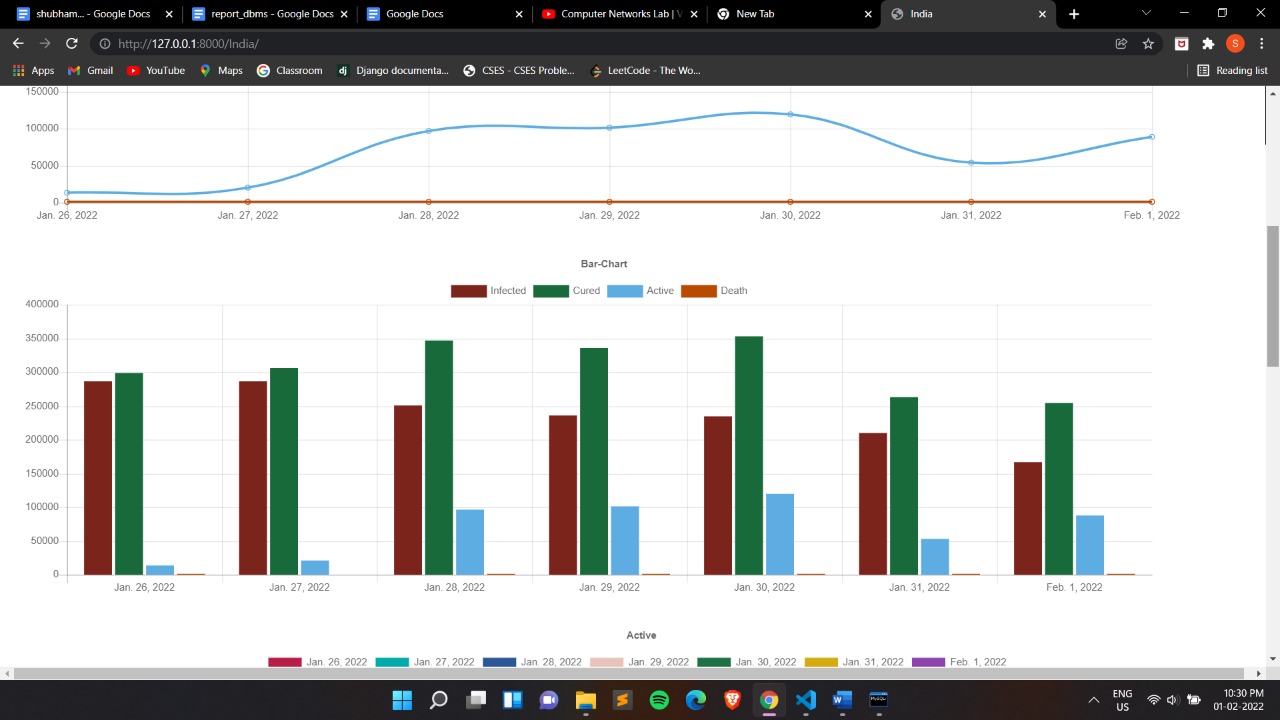


Figure 6.4: State bargraph



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 32 |

Covid-19 Dashboard 1BI19CS147



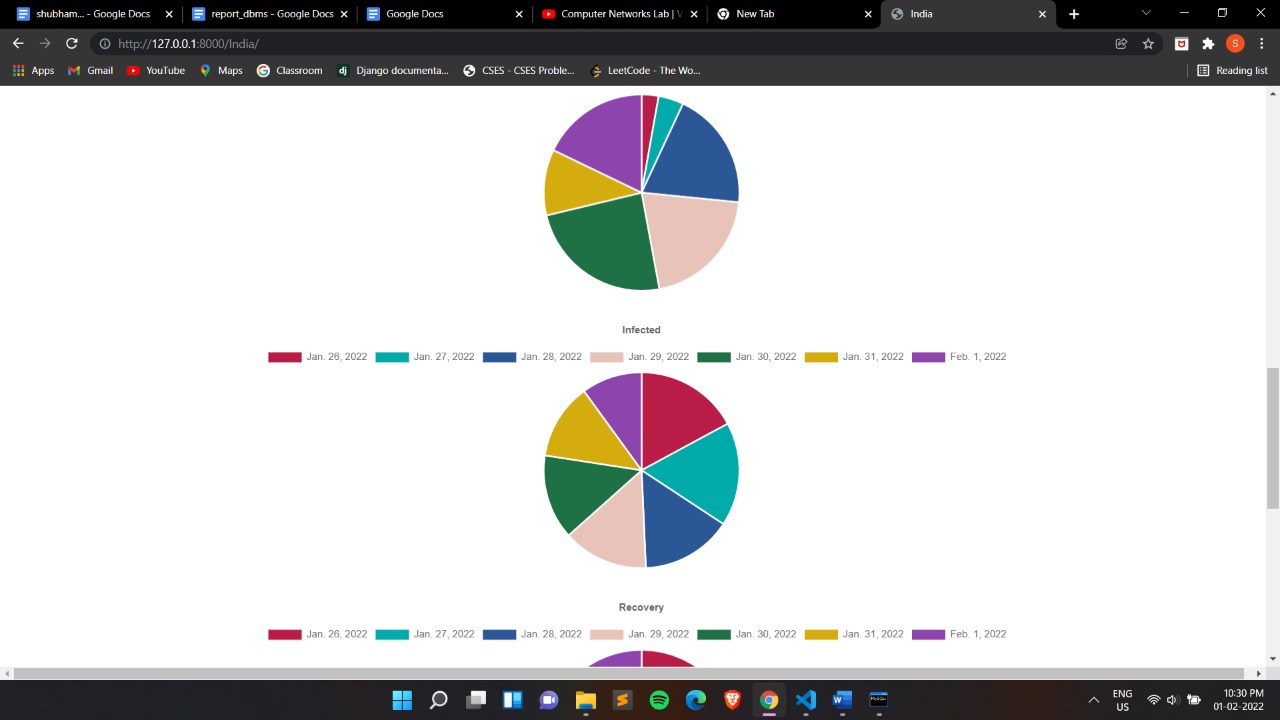


Figure 6.5: State piegraph

**6.3 Awareness page**

This page displays various awareness related to covid19 .

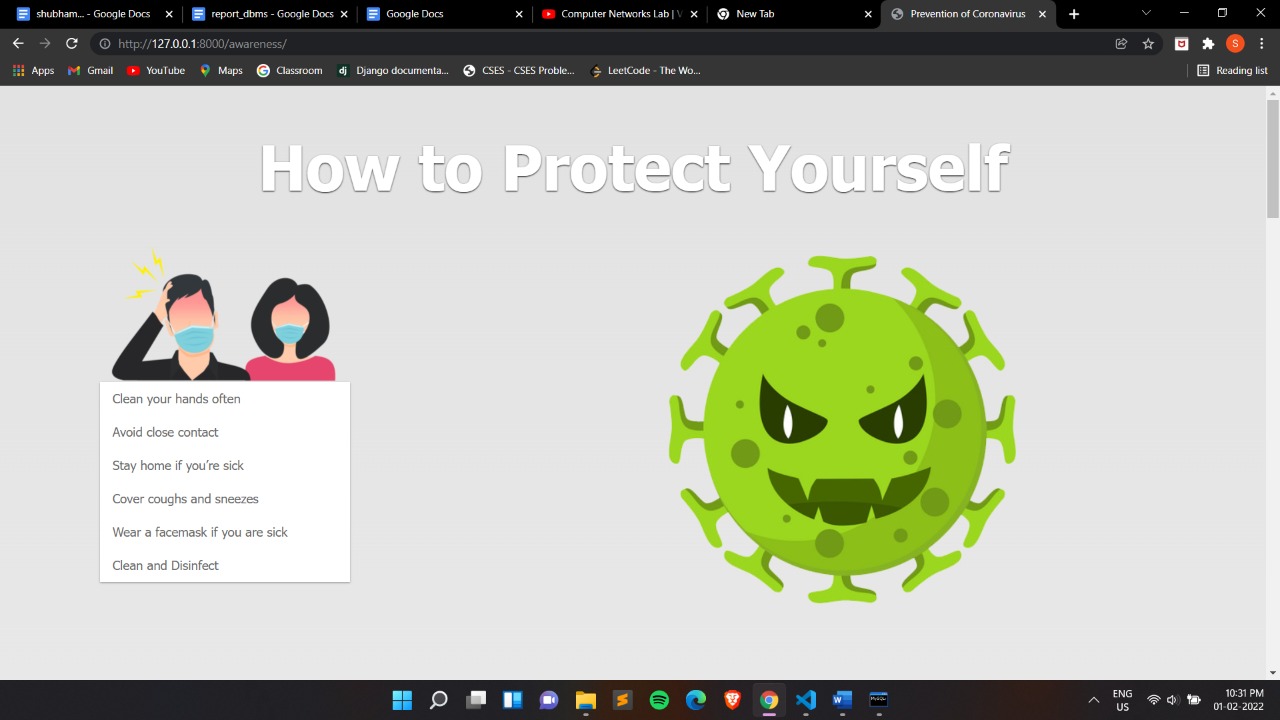


Figure 6.4: Measures



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 33 |

Covid-19 Dashboard 1BI19CS147



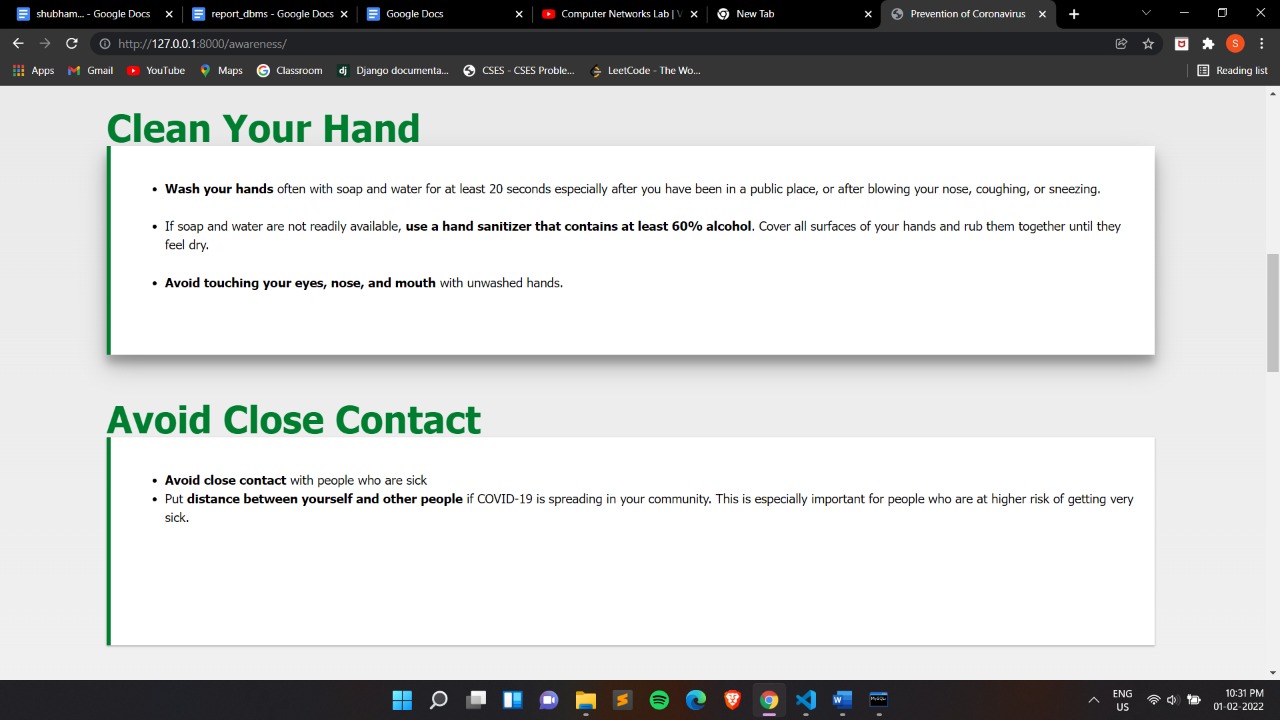


Figure 6.5: Measures2

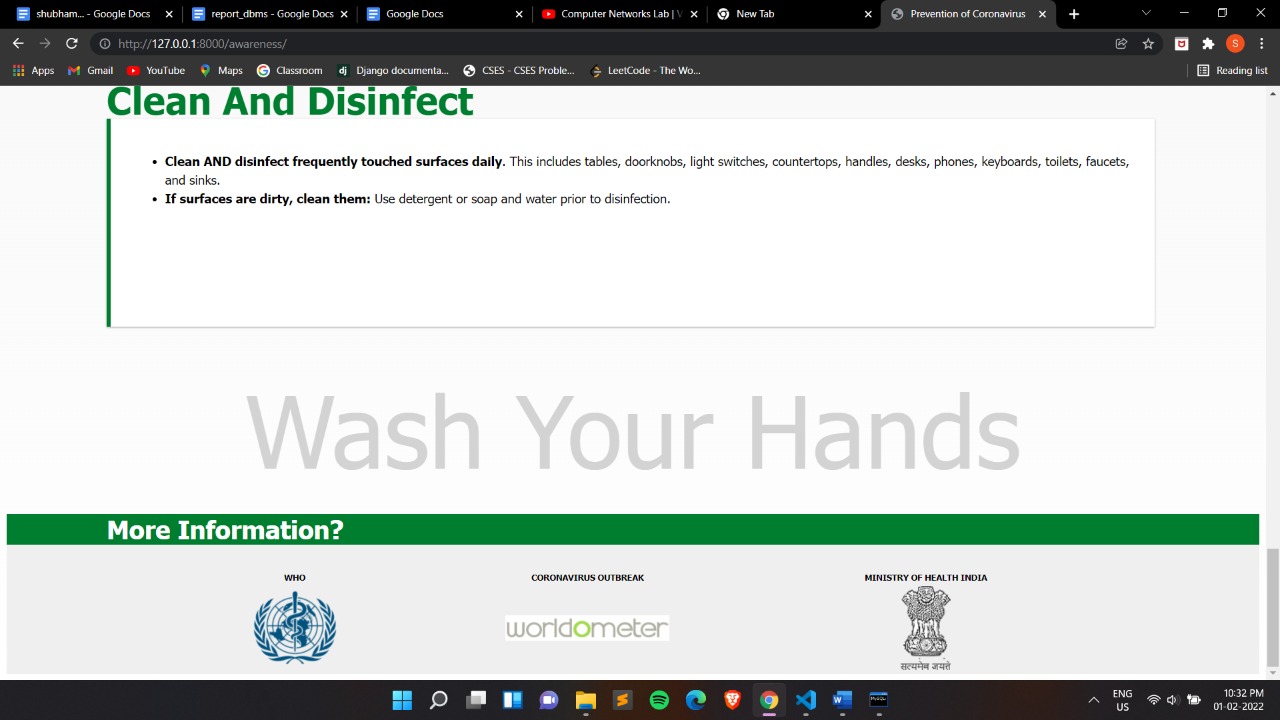


Figure 6.6: Measures3



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 34 |

**CHAPTER-7**

**APPLICATIONS**

**Application**

**Controlled governance with an end-to-end view reduces risk**

Many of the significant COVID-19 initiatives involve sensitive data, from patient data to personal banking data. There are great risks to data security posed by the swift influx of these new data sources into a company’s data ecosystem. Protecting customer data and excellence in data governance are primary priorities for a bank we work with, that have only grown in the time of coronavirus.

**Collaborative data features accelerate analytics**

The heightened need for faster analytics has been one of the salient priority shifts catalyzed by the coronavirus. Perhaps no sector has felt this pressure more than companies tasked with discovering treatments for the virus.

**ESG matters and will only grow in importance**

Both consumers and businesses were placing increased priority on environmental, social, and governance (ESG) scores, data, and analytics before COVID-19. The virus has heightened their importance. Companies that score well for ESG metrics, and those that provide the analytics around such scores, are consistently rated higher for trustworthiness and known as responsible investing leaders.



|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 35 |

**CHAPTER-8**

**CONCLUSION**

**CONCLUSION**

The volume of data increases dramatically over time, especially data generated on the global pandemic caused by COVID-19. Such a volume of data requires utilizing big data analytics tools along with AI techniques to make sense of the pandemic and control its spread in a timely manner. In this study, we presented a review of several data analysis applications for COVID-19, providing a taxonomy structure which classified the potential applications of COVID-19 into four categories, namely diagnosis, estimate or predict risk score, healthcare decision-making, and pharmaceutical. The paper introduced several data analysis tools and explained the main features of each tool. We also provided important insights on a number of challenges that might hinder the use of data analytics tools for COVID-19. These challenges include healthcare data security and patient privacy issues, the difficulty of sharing data with researchers, absence of data validation for some studies that may lead to biased results, and the patients’ cooperation in sharing part of their medical information. Finally, we highlighted and discussed a number of future directions that should be considered in further research and applications to assist stakeholders, such as governments, MoHs, hospitals, patients, and responsible authorities, to make decisions and predict the future.





|  |  |
| --- | --- |
| Department of CS&E, BIT 2021-22 | 36 |