VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

HEALTHCARE ANALYSIS USING DATA VISUALIZATION TOOLS

CSE3021 – Social and Information Networks a Project Report

Submitted to:

Dr Punitha K

Submitted by:

ABHINAV VIJAYAKUMAR (19BCE1311)

AMAR DIXIT (19BCE1875)

In partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

June 2021

TABLE OF CONTENTS

TABLE OF FIGURES

1. Abstract

The aim of the study was to analyze healthcare sector from the viewpoint of the planning, implementation and evaluation of the current situation. The purpose was to generate knowledge and implement it in a project that can be utilized in healthcare management.

Every medical practice and every hospital will at some point decide to incorporate a new technology, a new procedure, or a new building; hire a new doctor; or embark on literally hundreds of other projects that require going outside of the box and outside of the usual routine in order to bring the project to perfection. This presentation discusses project management for implementing better medical practice or a hospital management for the betterment of healthcare sector.

The purpose of the study is to assess the financing needs of healthcare infrastructure, and develop management model that identifies problems, a framework for implementation and helps to evaluate dynamically performance of healthcare infrastructure service in India.

Companies like Cigna use data visualization to illustrate an individual's health status compared with others in a similar demographic. Such illustrations can educate patients on areas where they need to improve, including cholesterol levels, body mass index, and exercise habits.

2. Introduction

Economic growth in a country largely depends on the standards of its social infrastructure. Healthcare is important areas of social infrastructure. It also covers care of the other healthcare organization objective of which can be met through healthcare infrastructure needs, management model that identifies problems, develops a framework for implementation and helps to evaluate dynamically healthcare infrastructure service performance and social security measures.

In the past decades there has been a succession of different approaches to the development of infrastructure for the healthcare services. There have been striking similarities among these approaches in both direction-and timing in many different countries, particularly in the developing world.

Through fields such as health data analysis and health informatics, medical organizations accrue large amounts of raw data. Examples of such data include patient diagnosis, outcome and length of stay.

It provides a way to share important findings with stakeholders who may not be data literate, such as hospital executives and administrators. Health data visualization can illuminate the general public and alert them to important developments in a crisis.

Visualizing health data is a powerful way to share urgent health information swiftly and effectively. Health data visualizations are helpful because they condense complex data into more straightforward presentations.

The Benefits of Using Data Visualization In Healthcare:

There are many benefits to using data visualization in health care. Here are a few ways data visualization can increase a hospital's efficiency.

Care Coordination

Simplifying patient care data through data visualization helps nurses interpret research and easily leverage their information. It also allows health care professionals to access the data of other patients with similar symptoms to understand common timelines for recovery.

Patient Education

Companies like Cigna use data visualization to illustrate an individual's health status compared with others in a similar demographic. Such illustrations can educate patients on areas where they need to improve, including cholesterol levels, body mass index, and exercise habits.

Public Health

Data visualizations are often used by the media to depict health-related trends on a large scale. An example could be confirmed COVID-19 cases in the U.S. by state. The visualization might show a graph of the U.S. with shading to illustrate which states have been affected most by the pandemic.

Operations

Operational data can help increase organizational transparency by identifying and measuring specific teams and individuals and gauging their performance against their respective metrics. This can help teams learn from each other, share effective practices, and improve the organization's operational strategy.

2.1 Motivation Behind the Project:

COVID-19 CRISIS IN INDIA:

There have been record numbers of cases and deaths

Case numbers and deaths in India are continuing to rise fast, fuelled by a new variant.

The country recorded the world's highest single-day total on Thursday.

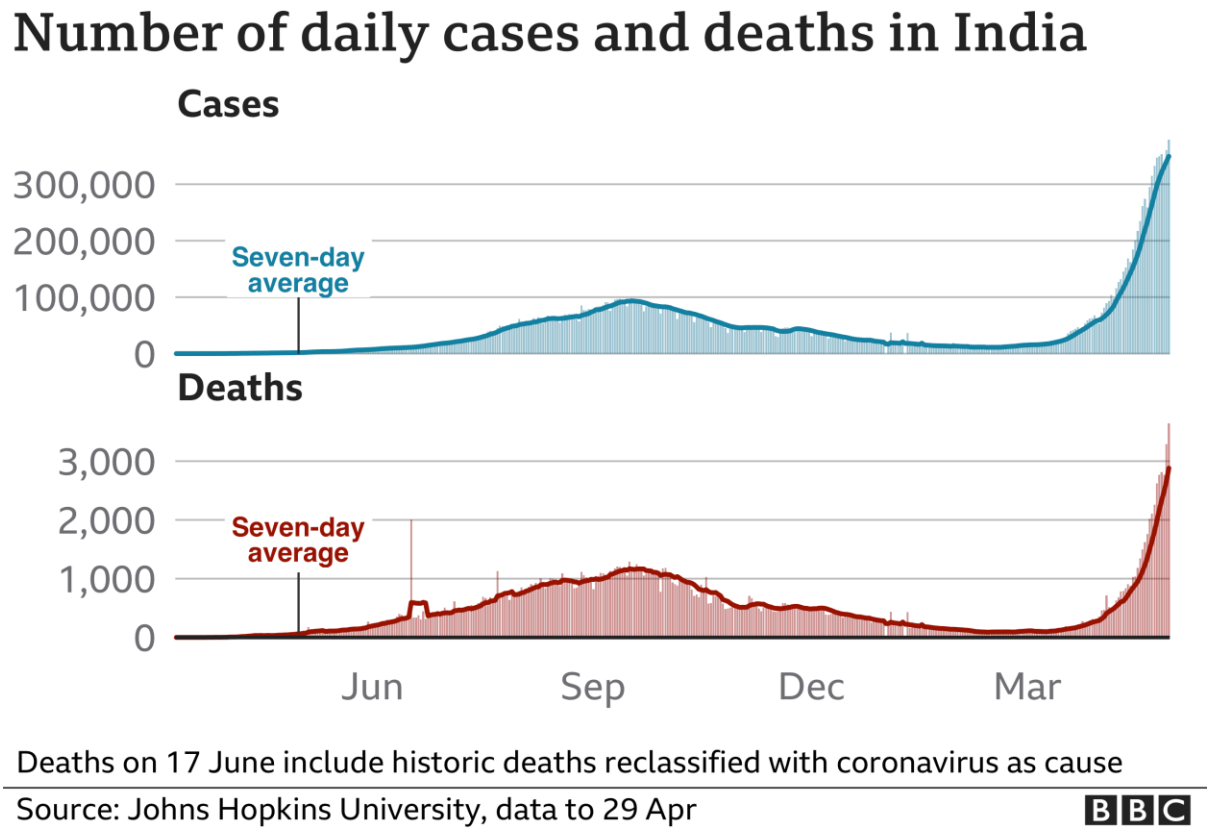


FIGURE 1

But the true numbers of cases and deaths are likely to be higher than the numbers provided by authorities, with many people avoiding testing or struggling to access it. Many deaths in rural areas also go unregistered.

Testing is increasing, but so too are the number of positive results.

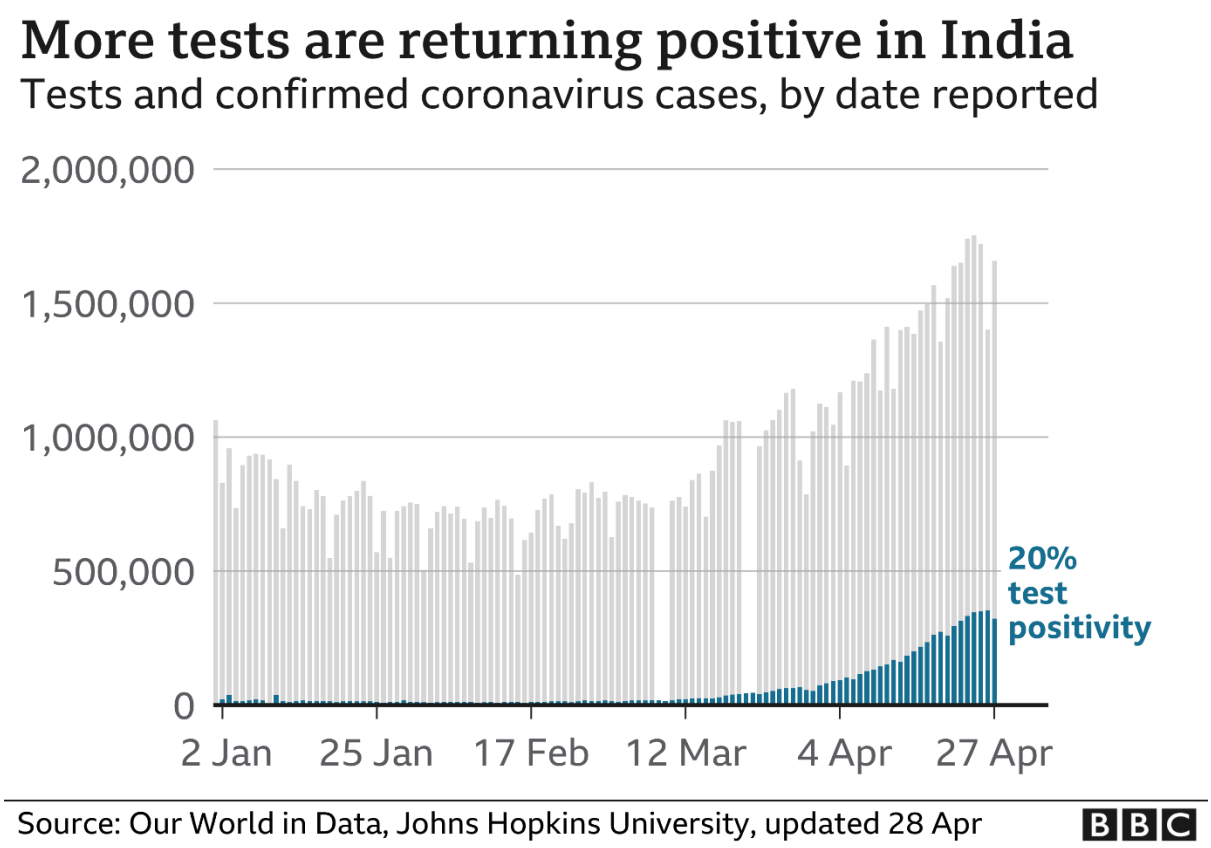


FIGURE 2

Last year, the World Health Organization recommended countries needed to get the positive test rate below 5% for at least two weeks before considering easing restrictions. The rate in India is now around 20%.

A high percentage of positive tests suggests high infection rates and the likelihood of many more people in the community with coronavirus undetected, according to Johns Hopkins University.

In total, India has confirmed more than 18 million infections and 200,000 deaths. Virologists say they expect the rate of infections to continue to increase for another two to three weeks.

- **Patients struggle at home as hospitals choke**
- **Why India is running out of oxygen again**
- **Countries send urgent aid to India**

There are very few critical care beds

The country has a chronic shortage of space on its intensive care wards, with many patients' families forced to drive for miles to try to find a bed for their loved one.

In Delhi - a region of about 20 million people - hospitals are full and are turning away new patients.

Doctors have described how people are dying on the streets outside hospitals as the country struggles to cope.

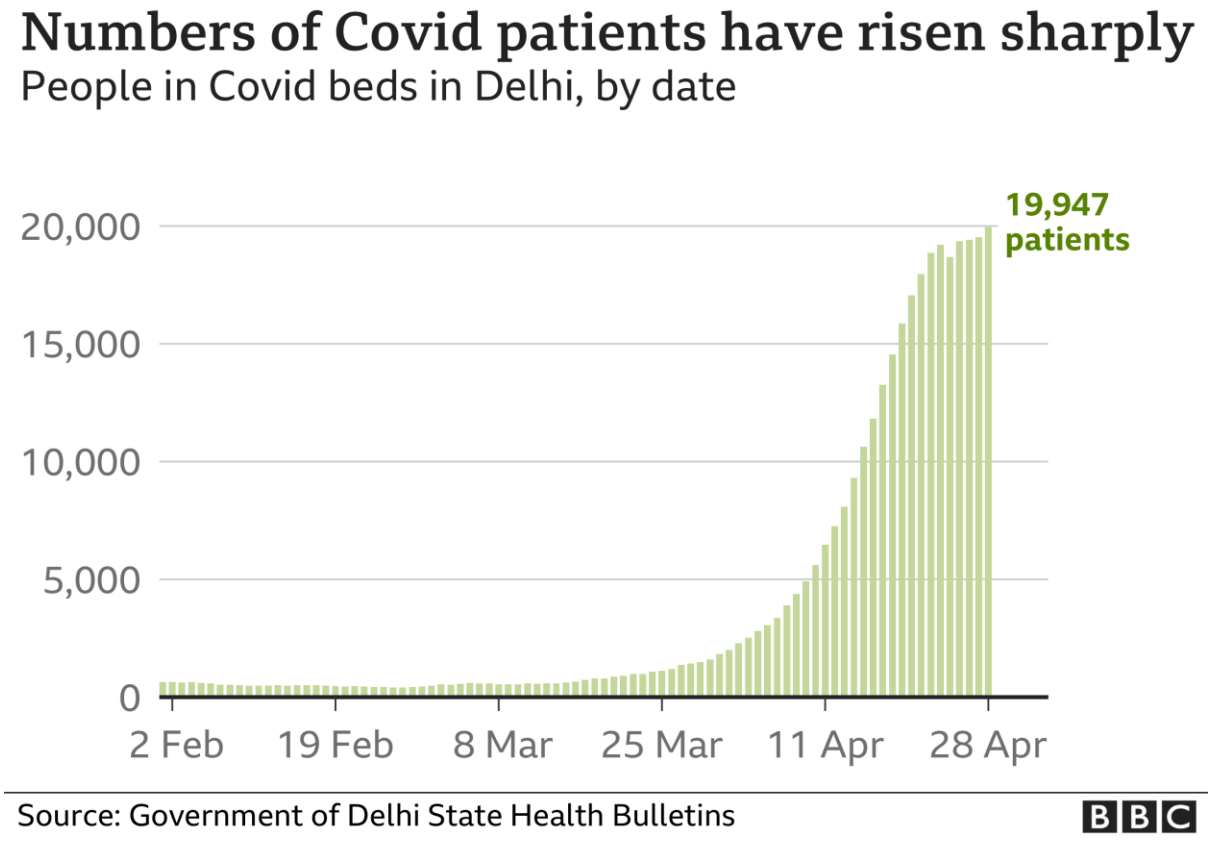


FIGURE 3

Some streets outside medical facilities have become crowded with the seriously ill, their loved ones trying to arrange stretchers and oxygen supplies for them as they plead with hospital authorities for a place inside.

"We have been roaming around for three days searching for a bed," one man told Reuters news agency as his wife sat immobile on the pavement.

FIGURE 4

IMAGE COPYRIGHTGETTY IMAGES

On Monday, the government announced military medical infrastructure would be made available to civilians and retired medical military personnel would be helping out in Covid health facilities.

There's a shortage of oxygen

Hospitals across India are also experiencing oxygen shortages, with some forced to put up signs warning of a lack of supplies.

The country now has the greatest demand for oxygen out of all other low, lower-middle and upper-middle-income countries, according to the PATH Oxygen Needs Tracker.

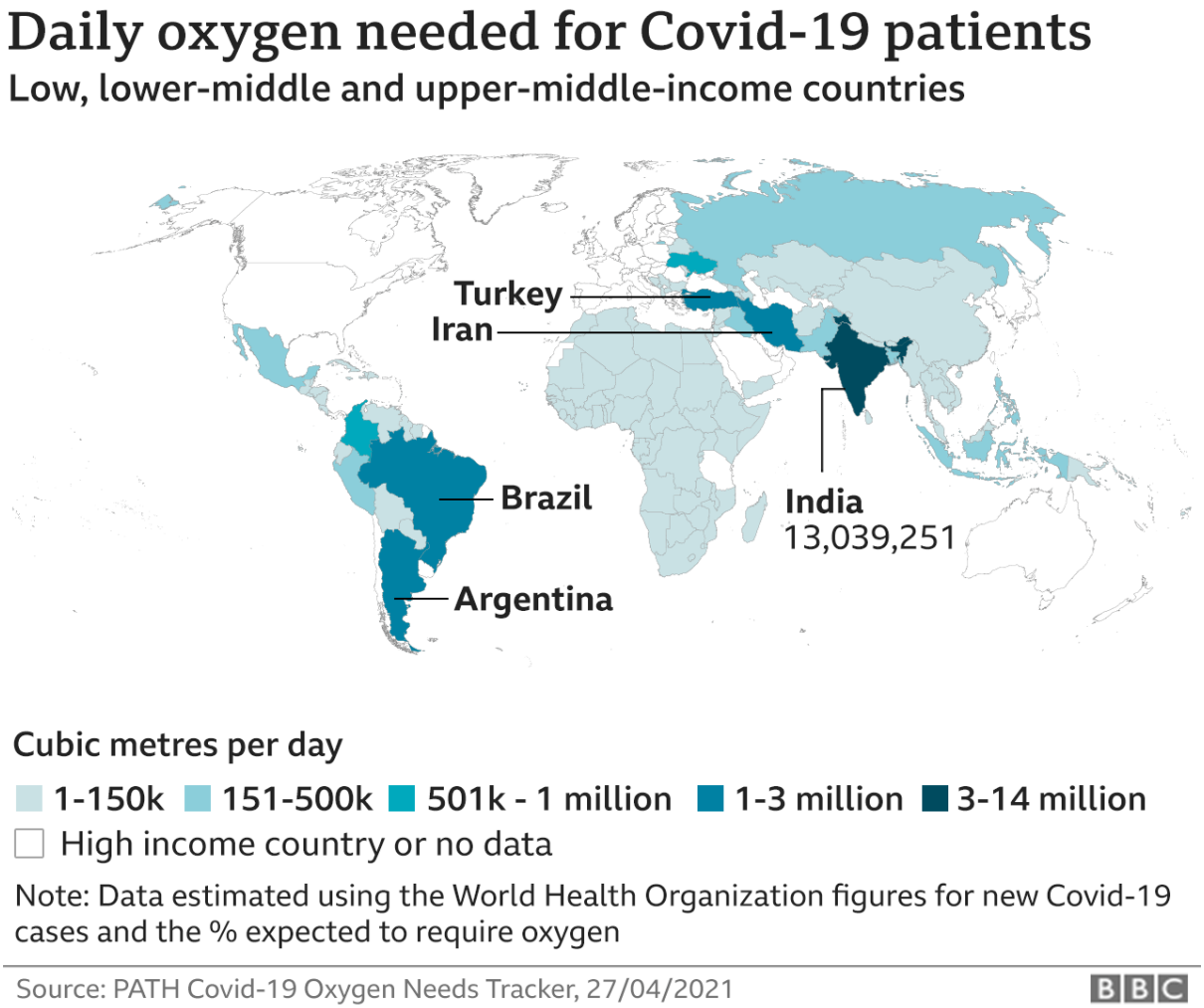


FIGURE 5

Demand has been growing between 6%-8% each day, according to PATH, an organisation that works with global institutions and businesses to tackle health problems.

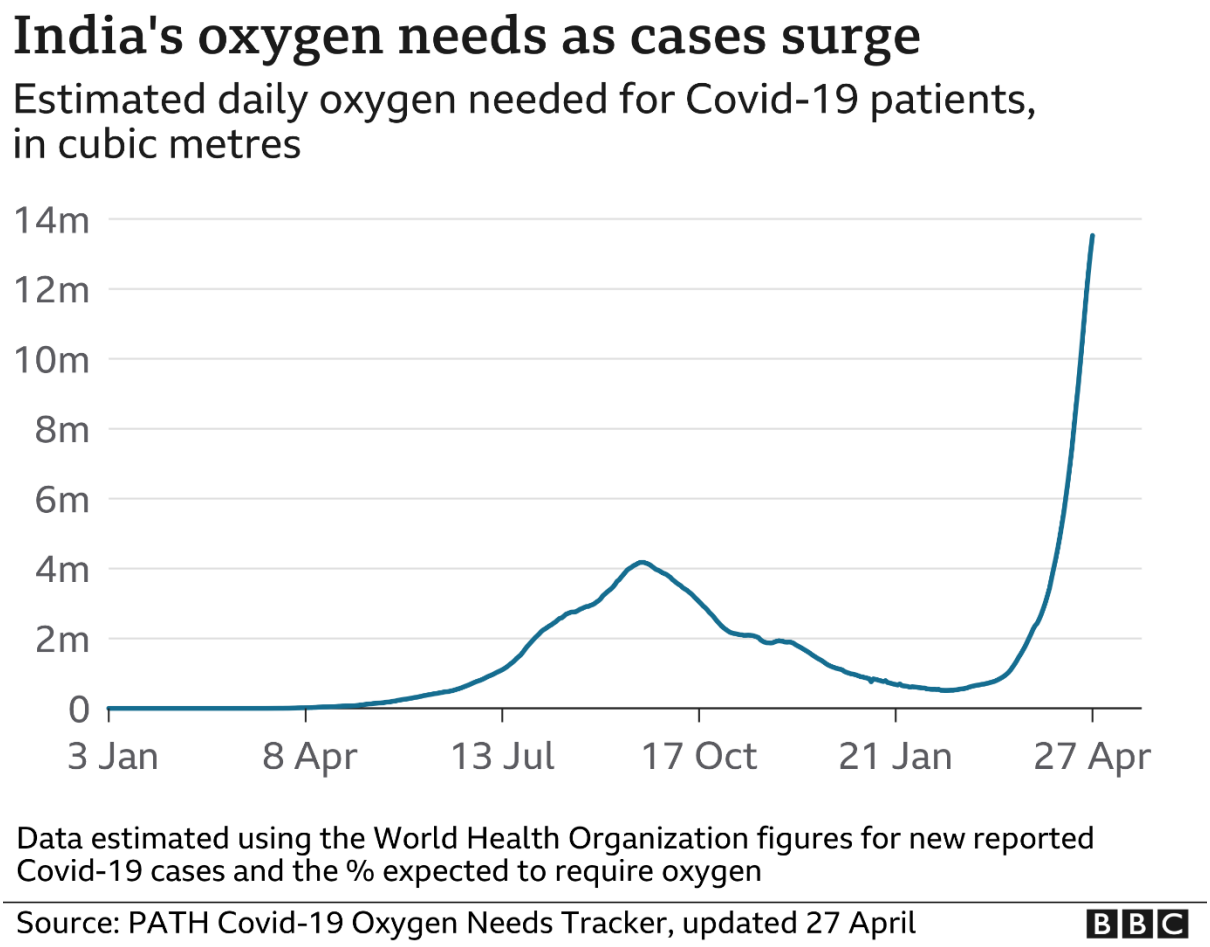

India's oxygen needs as cases surge
Estimated daily oxygen needed for Covid-19 patients, in cubic metres
14m
12m
10m
8m
6m
4m
2m
0
3 Jan
8 Apr
13 Jul
17 Oct
21 Jan
27 Apr
Data estimated using the World Health Organization figures for new reported Covid-19 cases and the % expected to require oxygen
Source: PATH Covid-19 Oxygen Needs Tracker, updated 27 April
BBC

FIGURE 6

3. Feasibility Study

Big data or population-based information has the potential to reduce uncertainty in medicine by informing clinicians about individual patient care. The objectives were: 1) to explore the feasibility of extracting and displaying population-based information from an actual clinical population's database records, 2) to explore specific design features for improving analysis display, 3) to explore perceptions of population information displays, and 4) to explore the impact of population information display on cognitive outcomes.

The qualitative data analysis for preferences of population information display resulted in four themes: 1) trusting the big/population data can be an issue, 2) embedded analytics is necessary to explore patient similarities, 3) need for tools to control the view (overview, zoom and filter), and 4) different presentations of the population display can be beneficial to improve the display.

A population database has great potential for reducing complexity and uncertainty in medicine to improve clinical care. The preferences identified for the population information display will guide future health information technology system designers for better and more intuitive display.

4. About the Dataset

Real time datasets were obtained from various reliable sources on the internet:

- Official websites maintained by the Govt.
- Kaggle.com

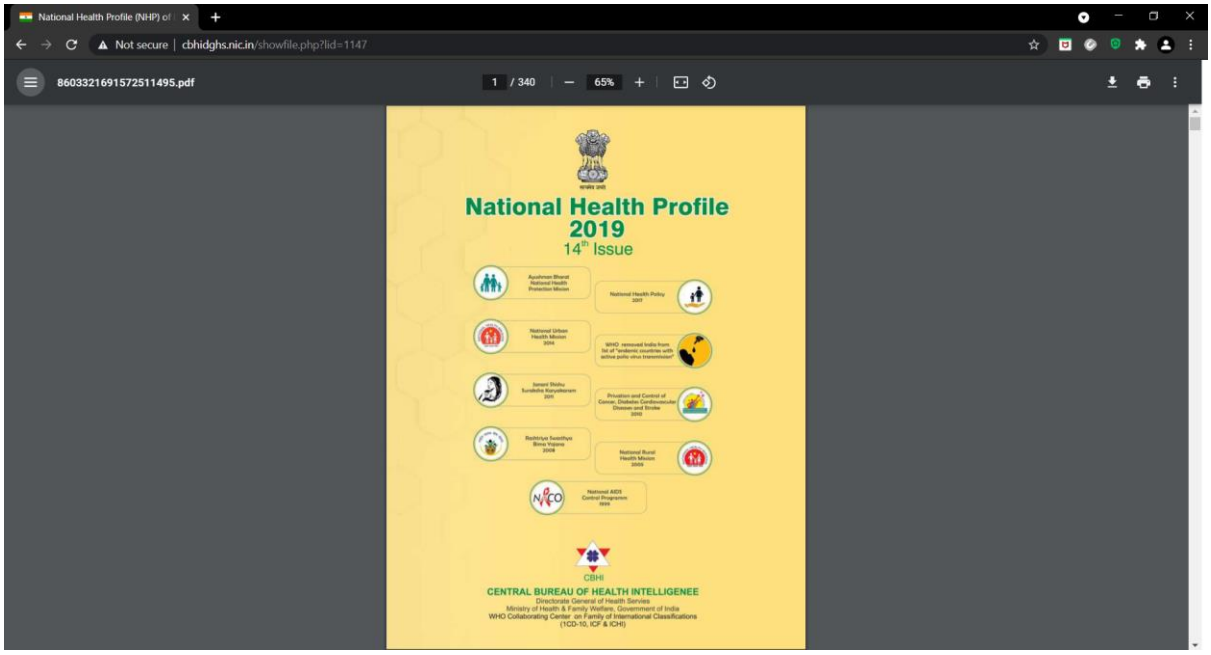
Kaggle.com

Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

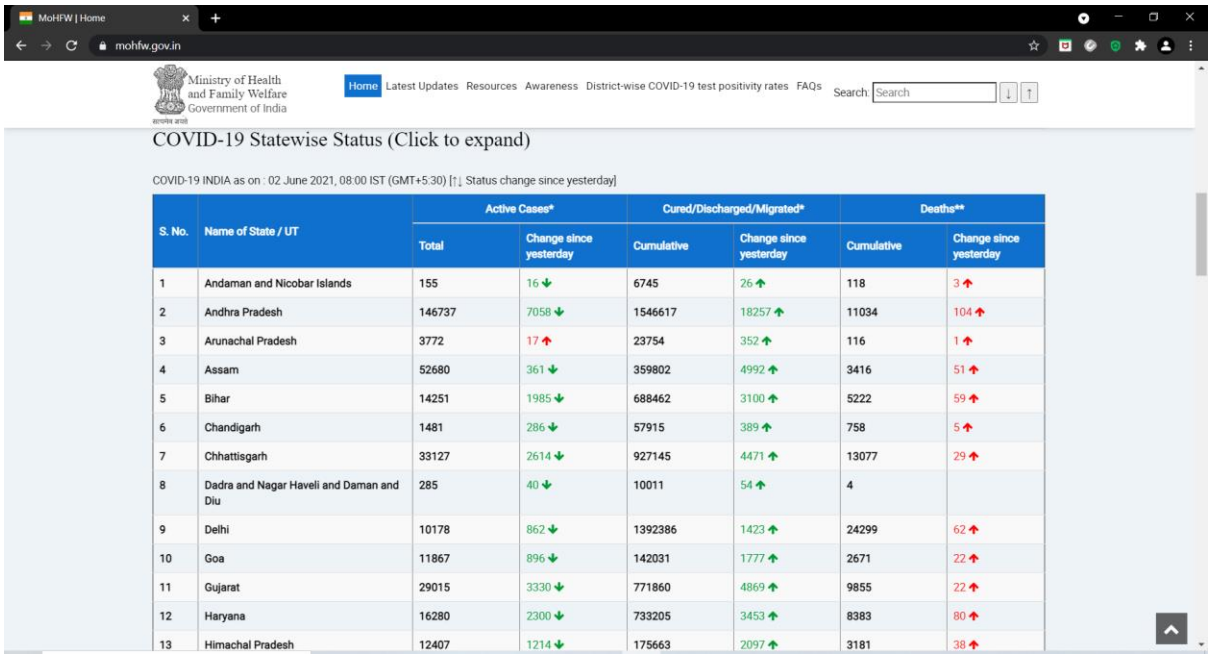
Kaggle got its start in 2010 by offering machine learning competitions and now also offers a public data platform, a cloud-based workbench for data science, and Artificial Intelligence education. Its key personnel were Anthony Goldbloom and Jeremy Howard. Nicholas Gruen was founding chair succeeded by Max Levchin. Equity was raised in 2011 valuing the company at $25 million. On 8 March 2017, Google announced that they were acquiring Kaggle.

Official websites maintained by the Govt:

National Health Profile:



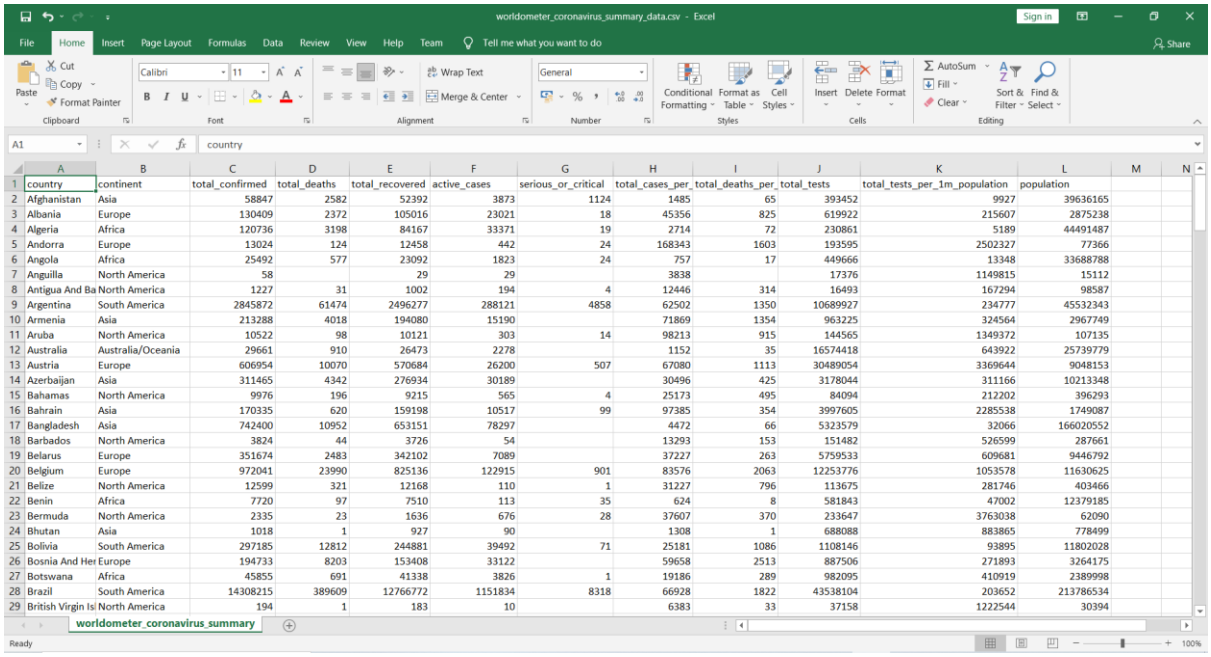
Ministry of Health and Family Welfare COVID-19 data:



S. No.	Name of State / UT	Active Cases*		Cured/Discharged/Migrated*		Deaths**	
		Total	Change since yesterday	Cumulative	Change since yesterday	Cumulative	Change since yesterday
1	Andaman and Nicobar Islands	155	16 ↓	6745	26 ↑	118	3 ↑
2	Andhra Pradesh	146737	7058 ↓	1546617	18257 ↑	11034	104 ↑
3	Arunachal Pradesh	3772	17 ↑	23754	352 ↑	116	1 ↑
4	Assam	52680	361 ↓	359802	4992 ↑	3416	51 ↑
5	Bihar	14251	1985 ↓	688462	3100 ↑	5222	59 ↑
6	Chandigarh	1481	286 ↓	57915	389 ↑	758	5 ↑
7	Chhattisgarh	33127	2614 ↓	927145	4471 ↑	13077	29 ↑
8	Dadra and Nagar Haveli and Daman and Diu	285	40 ↓	10011	54 ↑	4	
9	Delhi	10178	862 ↓	1392386	1423 ↑	24299	62 ↑
10	Goa	11867	896 ↓	142031	1777 ↑	2671	22 ↑
11	Gujarat	29015	3330 ↓	771860	4869 ↑	9855	22 ↑
12	Haryana	16280	2300 ↓	733205	3453 ↑	8383	80 ↑
13	Himachal Pradesh	12407	1214 ↓	175663	2097 ↑	3181	38 ↑

Datasets obtained from Kaggle:

World-wide Coronavirus Summary Dataset

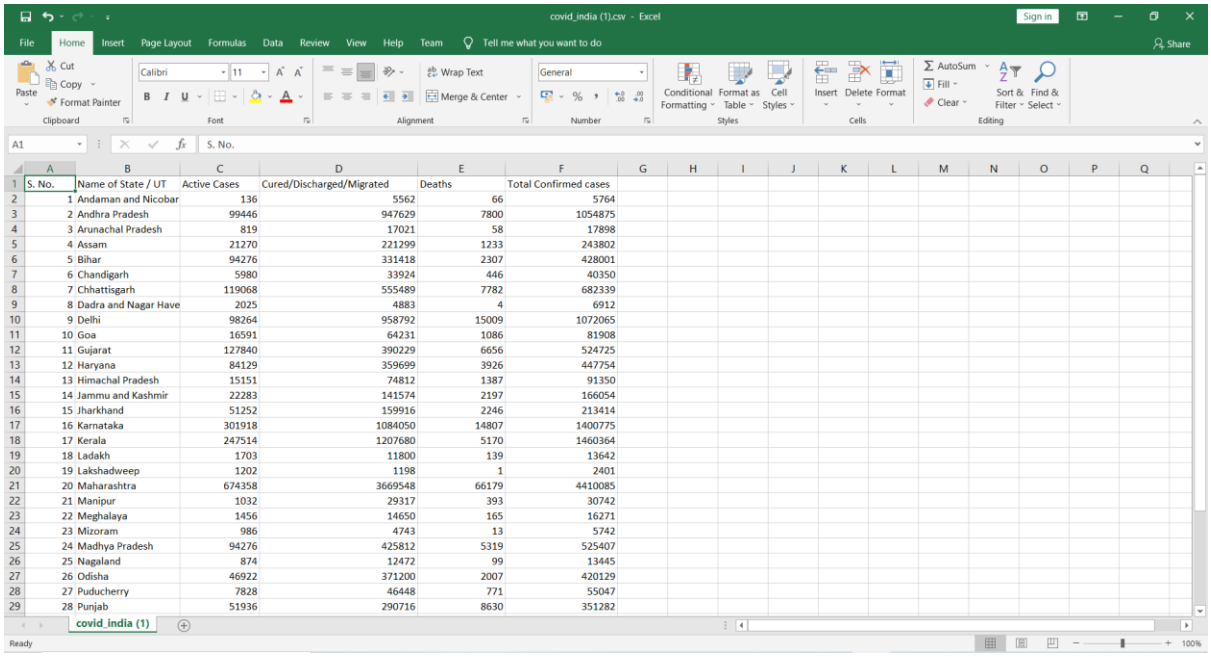


worldometer_coronavirus_summary_data.csv - Excel

country	continent	total_confirmed	total_deaths	total_recovered	active_cases	serious_or_critical	total_cases_per	total_deaths_per	total_tests	total_tests_per_1m_population	population
Afghanistan	Asia	58847	2582	52392	3873	1124	1485	65	393452	9927	39636165
Albania	Europe	130409	2372	105016	23021	18	45356	825	619922	215607	2875238
Algeria	Africa	120736	3198	84167	33371	19	2714	72	230861	5189	44491487
Andorra	Europe	13024	124	12458	442	24	168343	1603	193595	2502327	77366
Angola	Africa	25492	577	23092	1823	24	757	17	449666	13348	33688788
Anguilla	North America	58		29	29		3838		17376	1149815	15112
Antigua And Ba	North America	1227	31	1002	194	4	12446	314	16493	167294	98587
Argentina	South America	2845872	61474	2496277	288121	4858	62502	1350	10689927	234777	45532343
Armenia	Asia	213288	4018	194080	15190		71869	1354	963225	324564	2967749
Aruba	North America	10522	98	10121	303	14	98213	915	144565	1349372	107135
Australia	Australia/Oceania	29661	910	26473	2278		1152	35	16574418	643922	25739779
Austria	Europe	606954	10070	570684	26200	507	67080	1113	30489054	3369644	9048153
Azerbaijan	Asia	311465	4342	276934	30189		30496	425	3178044	311166	10213348
Bahamas	North America	9976	196	9215	565	4	25173	495	84094	212202	396293
Bahrain	Asia	170335	620	159198	10517	99	97385	354	3997605	2285538	1749087
Bangladesh	Asia	742400	10952	653151	78297		4472	66	5323579	32066	166020552
Barbados	North America	3824	44	3726	54		13293	153	151482	526599	287661
Belarus	Europe	351674	2483	342102	7089		37227	263	5759533	609681	9446792
Belgium	Europe	972041	23990	825136	122915	901	83576	2063	12253776	1053578	11630625
Belize	North America	12599	321	12168	110	1	31227	796	113675	281746	403466
Benin	Africa	7720	97	7510	113	35	624	8	581843	47002	12379185
Bermuda	North America	2335	23	1636	676	28	37607	370	233647	3763038	62090
Bhutan	Asia	1018	1	927	90		1308	1	688088	883865	778499
Bolivia	South America	297185	12812	244881	39492	71	25181	1086	1108146	93895	11802028
Bosnia And Her	Europe	194733	8203	153408	33122		59658	2513	887506	271893	3264175
Botswana	Africa	45855	691	41338	3826	1	19186	289	982095	410919	2389998
Brazil	South America	14308215	389609	12766772	1151834	8318	66928	1822	43538104	203652	213786534
British Virgin Is	North America	194	1	183	10		6383	33	37158	1222544	30394

worldometer_coronavirus_summary

India Covid-19 Dataset (State-wise):

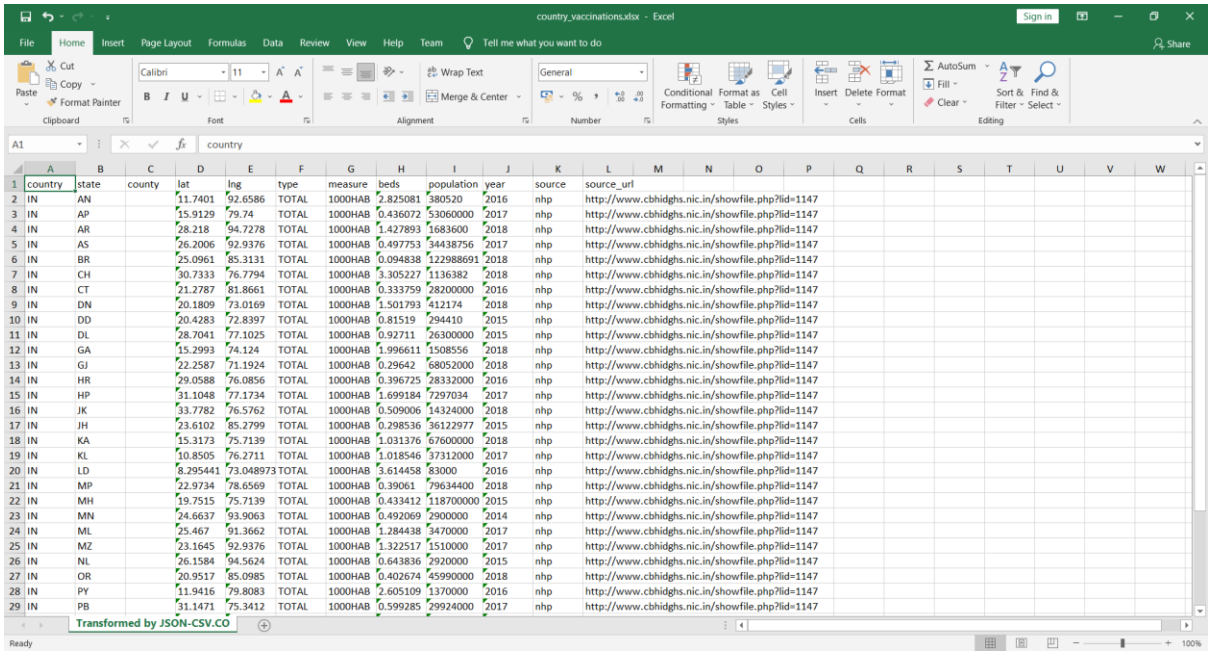


covid_india (1).csv - Excel

S. No.	Name of State / UT	Active Cases	Cured/Discharged/Migrated	Deaths	Total Confirmed cases
1	Andaman and Nicobar	136	5562	66	5764
2	Andhra Pradesh	99446	947629	7800	1054875
3	Arunachal Pradesh	819	17021	58	17898
4	Assam	21270	221299	1233	243802
5	Bihar	94276	331418	2307	428001
6	Chandigarh	5980	33924	446	40350
7	Chhattisgarh	119068	555489	7782	682339
8	Dadra and Nagar Have	2025	4883	4	6912
9	Delhi	98264	958792	15009	1072065
10	Goa	16591	64231	1086	81908
11	Gujarat	127840	390229	6656	524725
12	Haryana	84129	359699	3926	447754
13	Himachal Pradesh	15151	74812	1387	91350
14	Jammu and Kashmir	22283	141574	2197	166054
15	Jharkhand	51252	159916	2246	213414
16	Karnataka	301918	1084050	14807	1400775
17	Kerala	247514	1207680	5170	1460364
18	Ladakh	1703	11800	139	13642
19	Lakshadweep	1202	1198	1	2401
20	Maharashtra	674358	3669548	66179	4410085
21	Manipur	1032	29317	393	30742
22	Meghalaya	1456	14650	165	16271
23	Mizoram	986	4743	13	5742
24	Madhya Pradesh	94276	425812	5319	525407
25	Nagaland	874	12472	99	13445
26	Odisha	46922	371200	2007	420129
27	Puducherry	7828	46448	771	55047
28	Punjab	51936	290716	8630	351282

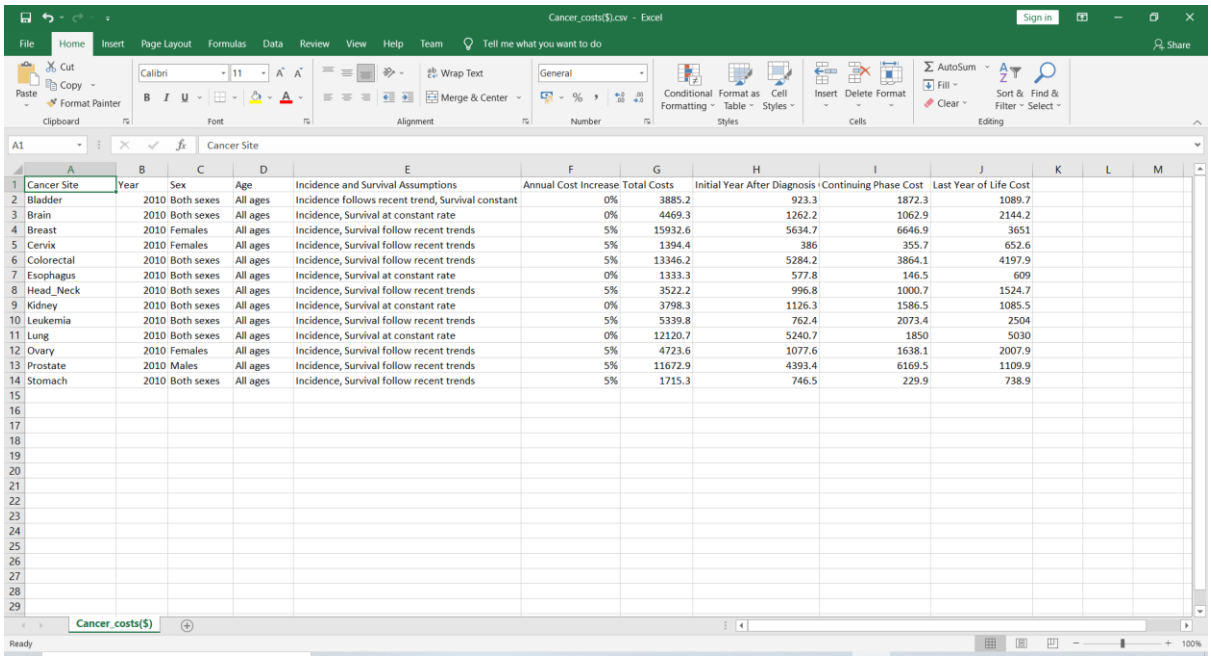
covid_india (1)

State-wise number of Beds in India:



country	state	county	lat	lng	type	measure	beds	population	year	source	source_url
IN	AN		11.7401	92.6586	TOTAL	1000HAB	2.825081	380520	2016	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	AP		15.9129	79.74	TOTAL	1000HAB	0.436072	53060000	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	AR		28.218	94.7278	TOTAL	1000HAB	1.427893	1683600	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	AS		26.2006	92.9376	TOTAL	1000HAB	0.497753	34438756	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	BR		25.0961	85.3131	TOTAL	1000HAB	0.094838	122988691	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	CH		30.7333	76.7794	TOTAL	1000HAB	3.305227	1136382	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	CT		21.2787	81.8661	TOTAL	1000HAB	0.333759	28200000	2016	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	DN		20.1809	73.0169	TOTAL	1000HAB	1.501793	412174	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	DD		20.4283	72.8397	TOTAL	1000HAB	0.81519	294410	2015	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	DL		28.7041	77.1025	TOTAL	1000HAB	0.92711	26300000	2015	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	GA		15.2993	74.124	TOTAL	1000HAB	1.996611	1508556	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	GJ		22.2587	71.1924	TOTAL	1000HAB	0.29642	68052000	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	HR		29.0588	76.0856	TOTAL	1000HAB	0.396725	28332000	2016	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	HP		31.1048	77.1734	TOTAL	1000HAB	1.699184	7297034	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	JK		33.7782	76.5762	TOTAL	1000HAB	0.509006	14324000	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	JH		23.6102	85.2799	TOTAL	1000HAB	0.298536	36122977	2015	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	KA		15.3173	75.7139	TOTAL	1000HAB	1.031376	67600000	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	KL		10.8505	76.2711	TOTAL	1000HAB	1.018546	37312000	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	LD		8.295441	73.048973	TOTAL	1000HAB	3.614458	83000	2016	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	MP		22.9734	78.6569	TOTAL	1000HAB	0.39061	79634400	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	MH		19.7515	75.7139	TOTAL	1000HAB	0.433412	118700000	2015	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	MN		24.6637	93.9063	TOTAL	1000HAB	0.492069	2900000	2014	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	ML		25.467	91.3662	TOTAL	1000HAB	1.284438	3470000	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	MZ		23.1645	92.9376	TOTAL	1000HAB	1.322517	1510000	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	NL		26.1584	94.5624	TOTAL	1000HAB	0.643836	2920000	2015	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	OR		20.9517	85.0985	TOTAL	1000HAB	0.402674	45990000	2018	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	PY		11.9416	79.8083	TOTAL	1000HAB	2.605109	1370000	2016	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147
IN	PB		31.1471	75.3412	TOTAL	1000HAB	0.599285	29924000	2017	nhp	http://www.cbhidghs.nic.in/showfile.php?lid=1147

Different type of Cancer Cost:



Cancer Site	Year	Sex	Age	Incidence and Survival Assumptions	Annual Cost Increase	Total Costs	Initial Year After Diagnosis	Continuing Phase Cost	Last Year of Life Cost
Bladder	2010	Both sexes	All ages	Incidence follows recent trend, Survival constant	0%	3885.2	923.3	1872.3	1089.7
Brain	2010	Both sexes	All ages	Incidence, Survival at constant rate	0%	4469.3	1262.2	1062.9	2144.2
Breast	2010	Females	All ages	Incidence, Survival follow recent trends	5%	15932.6	5634.7	6646.9	3651
Cervix	2010	Females	All ages	Incidence, Survival follow recent trends	5%	1394.4	386	355.7	652.6
Colorectal	2010	Both sexes	All ages	Incidence, Survival follow recent trends	5%	13346.2	5284.2	3864.1	4197.9
Esophagus	2010	Both sexes	All ages	Incidence, Survival at constant rate	0%	1333.3	577.8	146.5	609
Head_Neck	2010	Both sexes	All ages	Incidence, Survival follow recent trends	5%	3522.2	996.8	1000.7	1524.7
Kidney	2010	Both sexes	All ages	Incidence, Survival at constant rate	0%	3798.3	1126.3	1586.5	1085.5
Leukemia	2010	Both sexes	All ages	Incidence, Survival follow recent trends	5%	5339.8	762.4	2073.4	2504
Lung	2010	Both sexes	All ages	Incidence, Survival at constant rate	0%	12120.7	5240.7	1850	5030
Ovary	2010	Females	All ages	Incidence, Survival follow recent trends	5%	4723.6	1077.6	1638.1	2007.9
Prostate	2010	Males	All ages	Incidence, Survival follow recent trends	5%	11672.9	4393.4	6169.5	1109.9
Stomach	2010	Both sexes	All ages	Incidence, Survival follow recent trends	5%	1715.3	746.5	229.9	738.9

5. Design and Flow of Modules:

Following are the modules in which we have divided our project:

Patient hospital record modules:

- Patient appointment booking
- Patient medical record check
- Appointment booking System

Statistics based modules:

- Analysis of road accidents in India in the year 2020.
- Analysis of bed availability in hospitals across India during COVID-19 pandemic.
- COVID-19
    - Total cases worldwide
    - Total recovery worldwide
    - Total confirmed cases state wise in India
    - Total deaths state wise in India
- Cancer treatment costs

The above flow list of modules is also followed throughout the project implementation as well as in this project report.

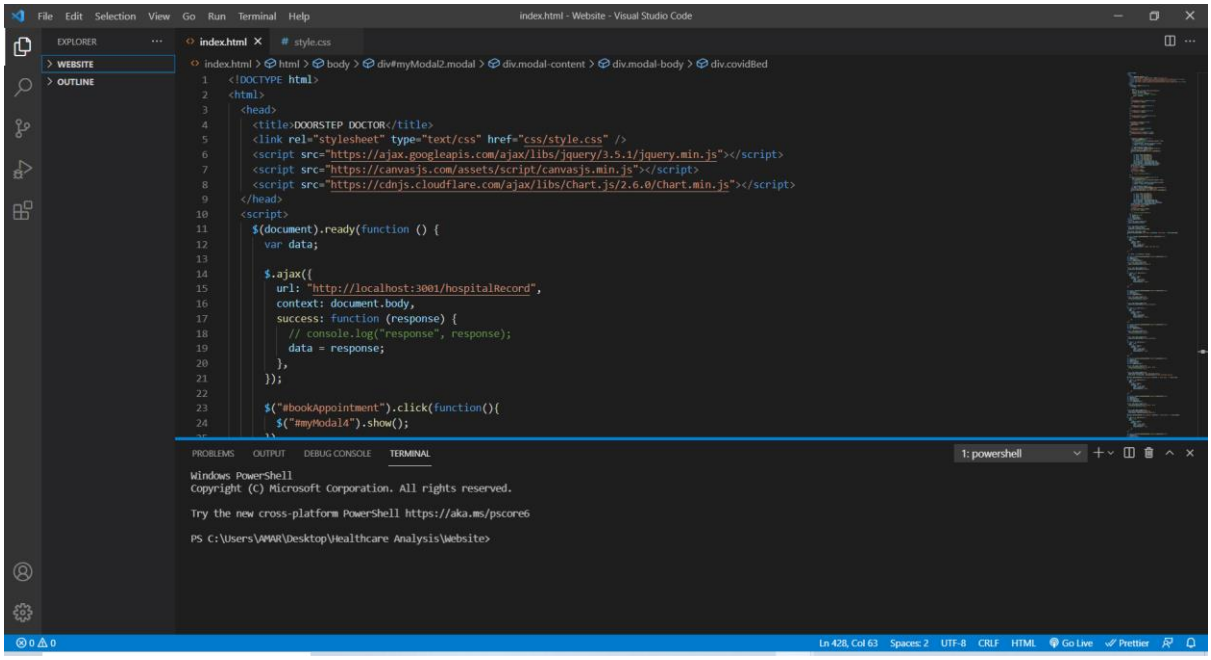
6. Module List

- **Implementation of basic features through webpage**
- **Analysis of road accidents in India in the year 2020.**
- **Analysis of bed availability in hospitals across India during COVID-19 pandemic.**
- **Patient appointment booking**
- **Patient medical record check**
- **Appointment booking System**
- **COVID-19**
    - ❖ **Total cases worldwide**
    - ❖ **Total recovery worldwide**
    - ❖ **Total confirmed cases state wise in India**
    - ❖ **Total deaths state wise in India**
- **Cancer treatment costs**

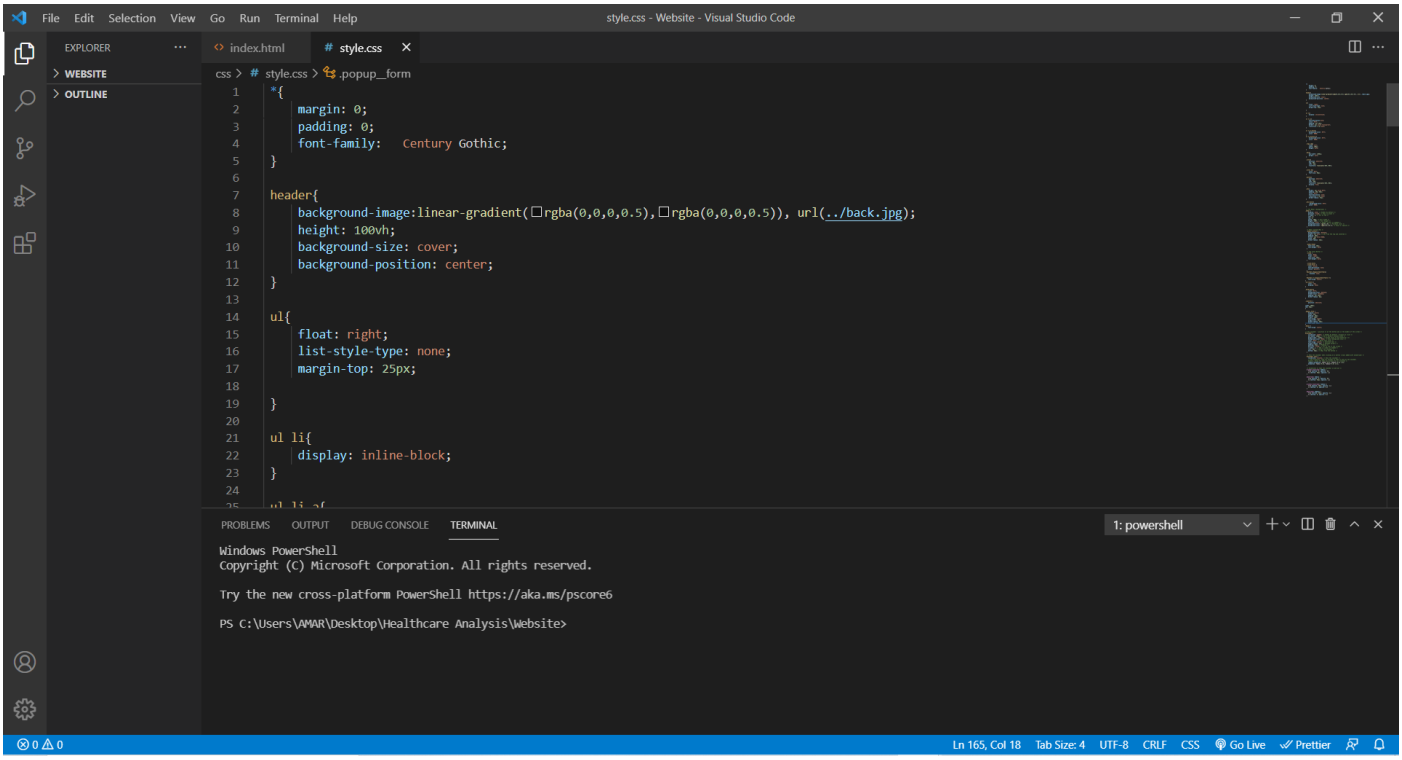
6.1 UID and Front-end validations:

6.1.1- Front-end code:

HTML and JavaScript Code:



CSS Code:



6.2 Web-based application:

The name of the Web-based application is "DOORSTEP DOCTOR".

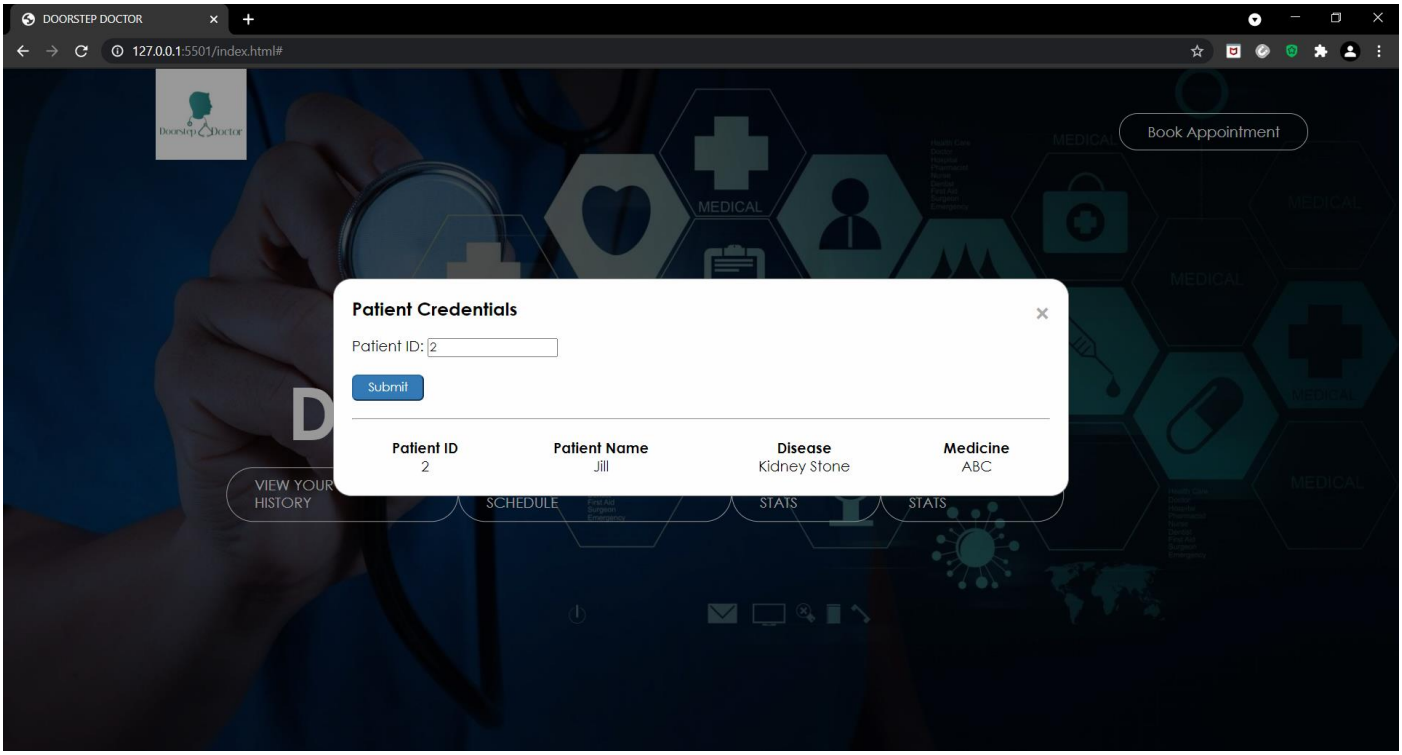
6.3- View your Medical History Module:

Code:

```
<!-- The Medical History Modal -->
<div id="myModal" class="modal">
  <!-- Modal content -->
  <div class="modal-content">
    <div class="modal-head">
      <span class="close">&times;</span>
      <p>Patient Credentials</p>
    </div>
    <br />
    <div class="modal-body">
      <form id="myForm" action="">
        <label for="fname">Patient ID: </label>
        <input type="number" id="fname" name="fname" /><br /><br />
        <input type="submit" value="Submit" class="btnPrimary"/>
      </form>
      <br />
      <hr />
      <br />
      <div id="record">
        <table style="width: 100%" id="myTable">
          <thead>
            <tr>
              <th>Patient ID</th>
              <th>Patient Name</th>
              <th>Disease</th>
              <th>Medicine</th>
            </tr>
          </thead>
  </table>
  <div class="no-record">No Record Found</div>
      </div>
    </div>
</div>
```

Patient need to click on **VIEW YOUR MEDICAL HISTORY** button

On entering his **Patient Id**, he/she can view their medical history.



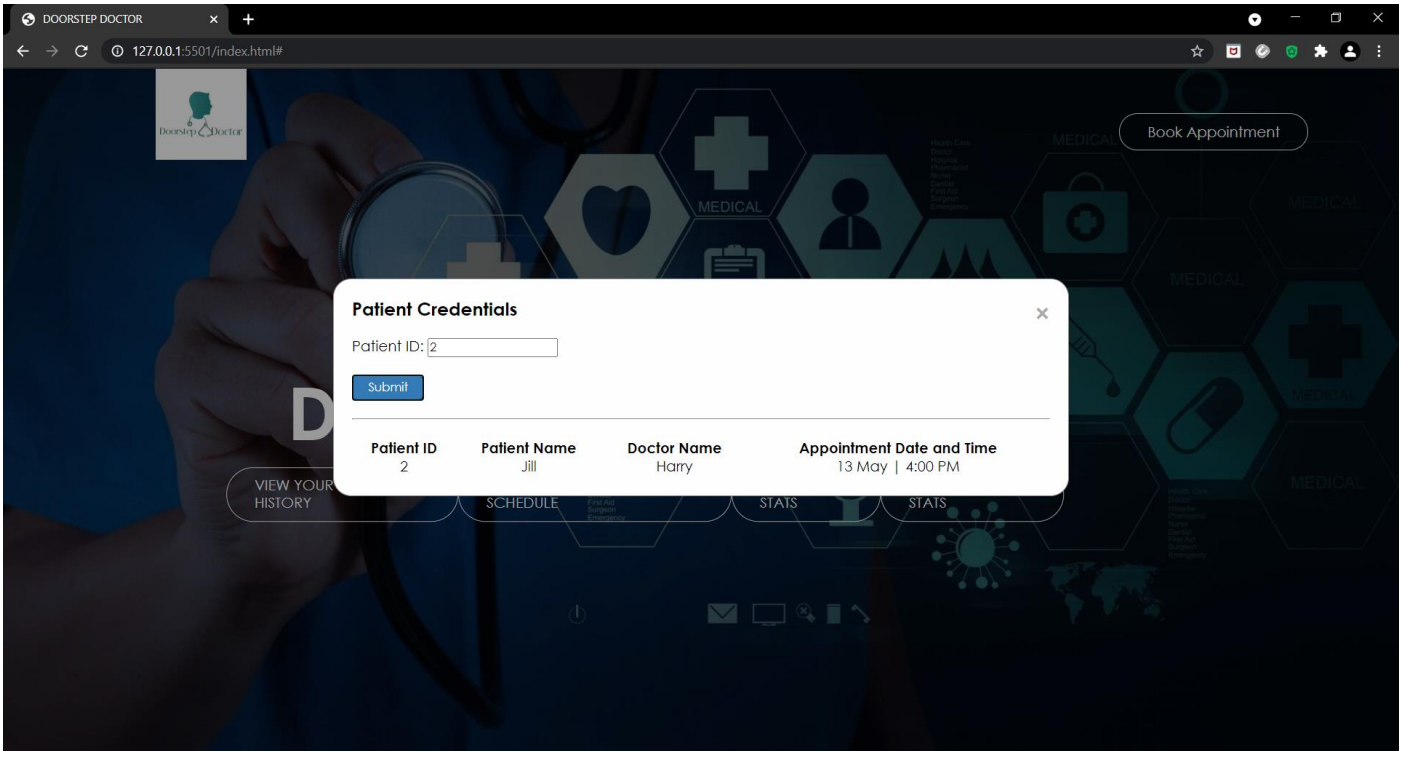
6.4- View Appointment Schedule Module:

Code:

```
<!-- The Appointment History Modal -->
<div id="appointmentHistoryModal" class="modal">
  <!-- Modal content -->
  <div class="modal-content">
    <div class="modal-head">
    <span class="close">&times;</span>
    <p>Patient Credentials</p>
    </div>
    <br />
    <div class="modal-body">
    <form id="appointHistoryForm" action="">
      <label for="fname">Patient ID: </label>
      <input type="number" id="fname" name="fname" /><br /><br />
      <input type="submit" value="Submit"  class="btnPrimary" />
    </form>
    <br />
    <hr />
    <br />
    <div id="record">
      <table style="width: 100%" id="appointmentTable">
        <thead>
          <tr>
          <th>Patient ID</th>
          <th>Patient Name</th>
          <th>Doctor Name</th>
          <th>Appointment Date and Time</th>
          </tr>
        </thead>
        </table>
      <div class="no-record">No Record Found</div>
    </div>
    </div>
  </div>
```

Patient need to click on **VIEW YOUR APPOINTMENT HISTORY** button

On entering his **Patient Id**, he/she can view their appointment schedule.



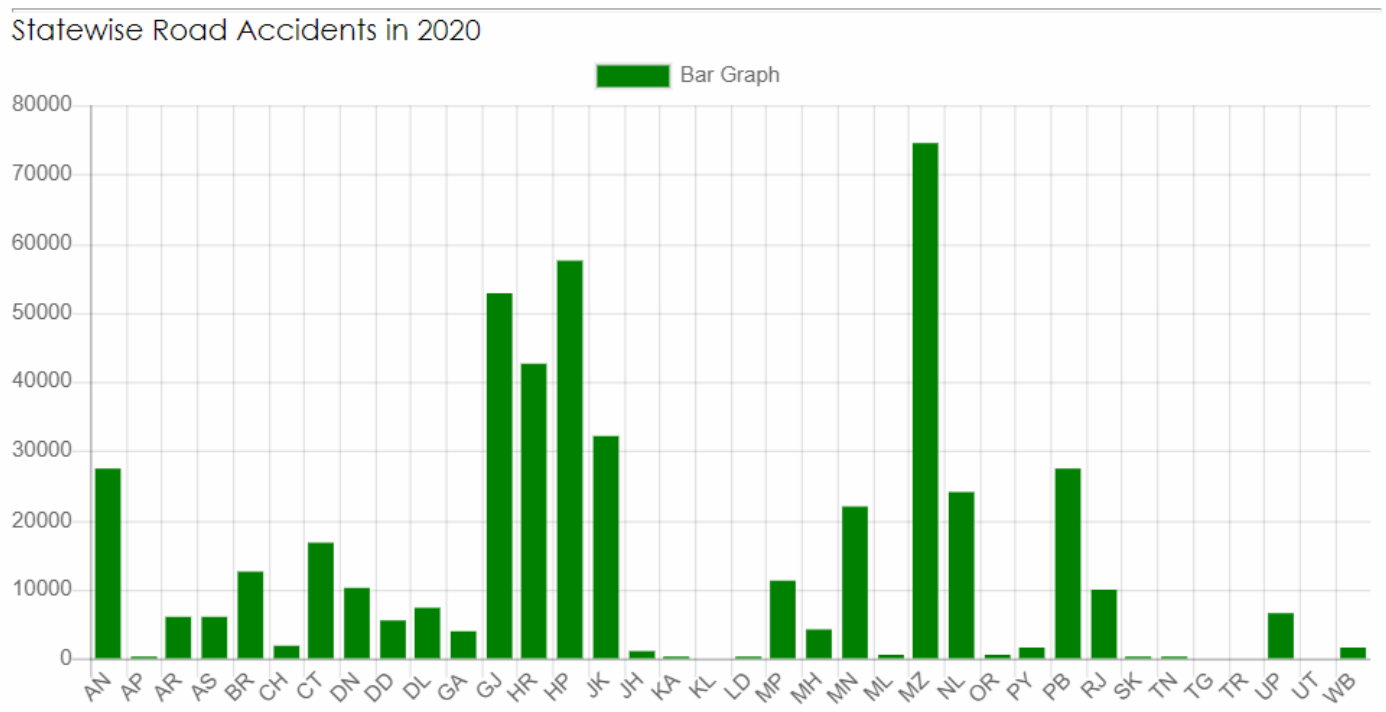
6.5 Road accidents in India in the year 2020.

Code:

```
var ctx_2 = document.getElementById("canvas1").getContext('2d');
var labels1=[];
var dataPoints1=[];
function addData1(data) {
  // console.log("data2",data);
for(var i=0;i<data.length;i++){
  labels1.push(data[i]['States/UTs'])
}

for(var j=0;j<data.length;j++){
  dataPoints1.push(data[j]['2020'])
}
}
var chart1 = new Chart(ctx_2, {
    type: 'bar',
    data: {
        labels: labels,
       datasets: [{
            label: 'Bar Graph',
            data: dataPoints1,
            backgroundColor: 'green',

       }]
    }
});
```

User has to click on **COUNTRY STATS** button to view country statistics.

Graph:

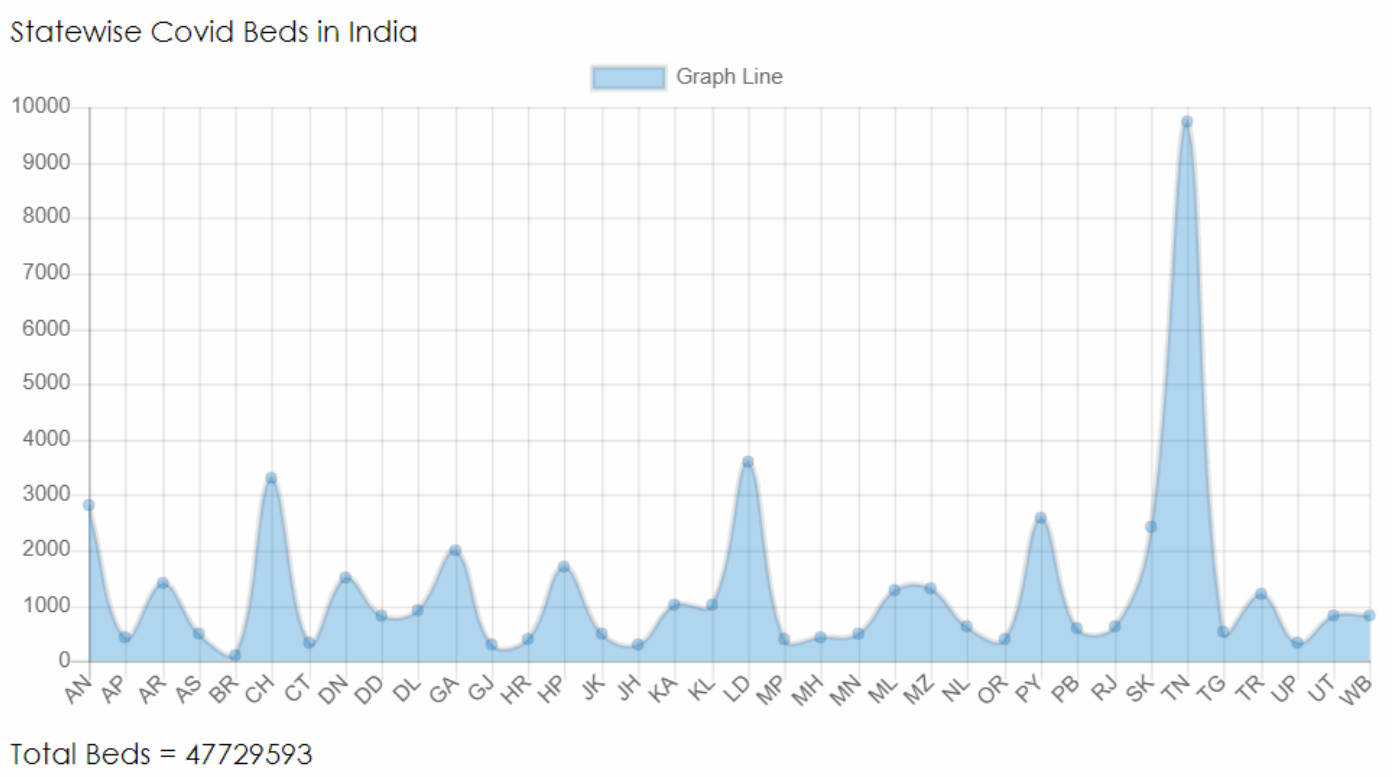


6.6 Bed availability in hospitals across India during COVID-19 pandemic.

Code:

```
    var dataPoints = [];
    var labels =[];
    var total_beds=0;
function addData(data) {
  debugger
for(var i=0;i<data.length;i++){
  labels.push(data[i].state)
}

for(var j=0;j<data.length;j++){
  total_beds +=data[j].beds;
  dataPoints.push(data[j].beds*1000)
}
 total_beds =total_beds *1000;
document.getElementById('total_beds').innerText= "Total Beds = "+total_beds*1000;
}

var ctx = document.getElementById("canvas").getContext('2d');
var config = {
   type: 'line',
   data: {
       labels: labels,
      datasets: [{
          label: 'Graph Line',
           data: dataPoints,
           backgroundColor: 'rgba(0, 119, 204, 0.3)',

      }]
   }
};
```

Graph:

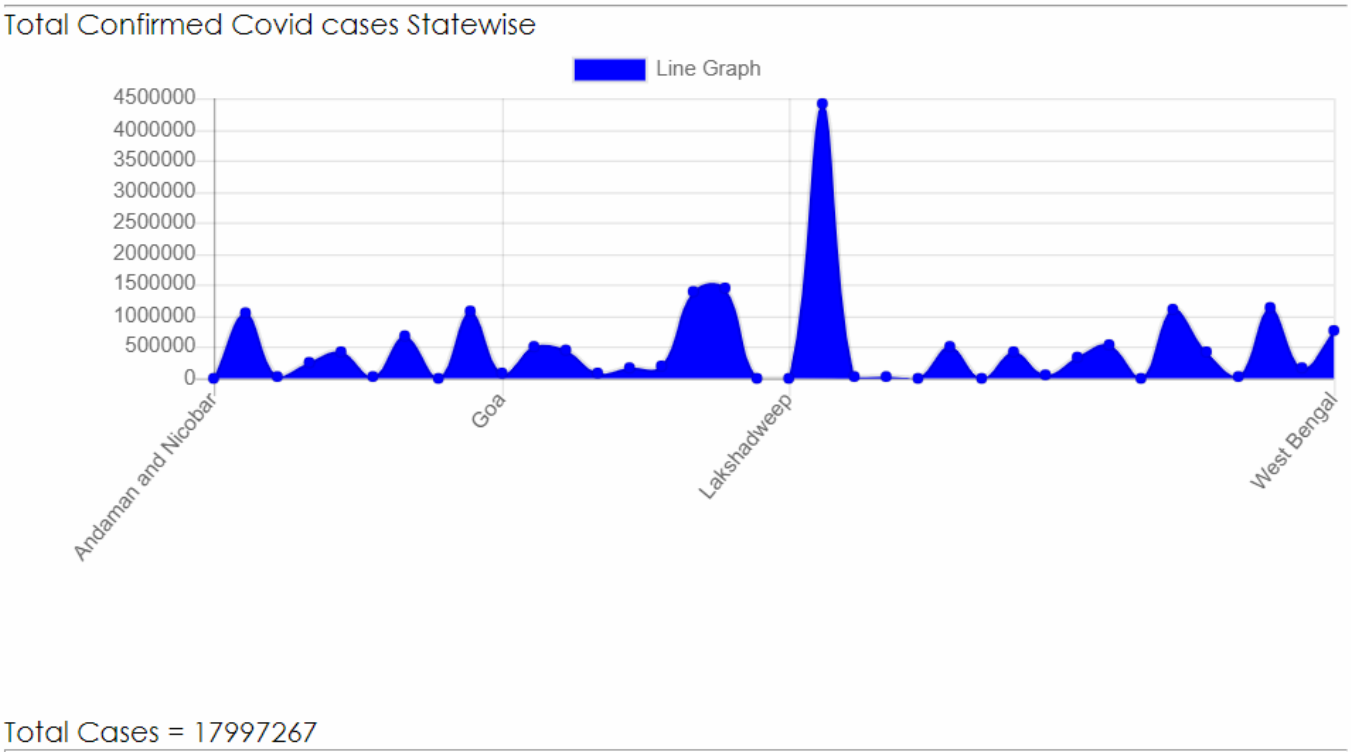


6.7 Total COVID-19 confirmed cases state wise in India

Code:

```
var ctx_5 = document.getElementById("canvas4").getContext('2d');
var labels4=[];
var dataPoints4=[];
var total_cases=0;
function addData5(data) {

for(var i=0;i<data.length;i++){
  labels4.push(data[i]["Name of State / UT"])
  // console.log("data32",data);
}

for(var j=0;j<data.length;j++){
  dataPoints4.push(data[j]['Total Confirmed cases'])
  total_cases = total_cases + parseInt(data[j]['Total Confirmed cases']);
}
document.getElementById("total_cases").innerText = 'Total Cases = '+ total_cases;
}
var chart = new Chart(ctx_5, {
    type: 'line',
    data: {
        labels: labels4,
        datasets: [{
            label: 'Line Graph',
            data: dataPoints4,
            backgroundColor: 'blue',

        }]
    }
});
```

Graph:

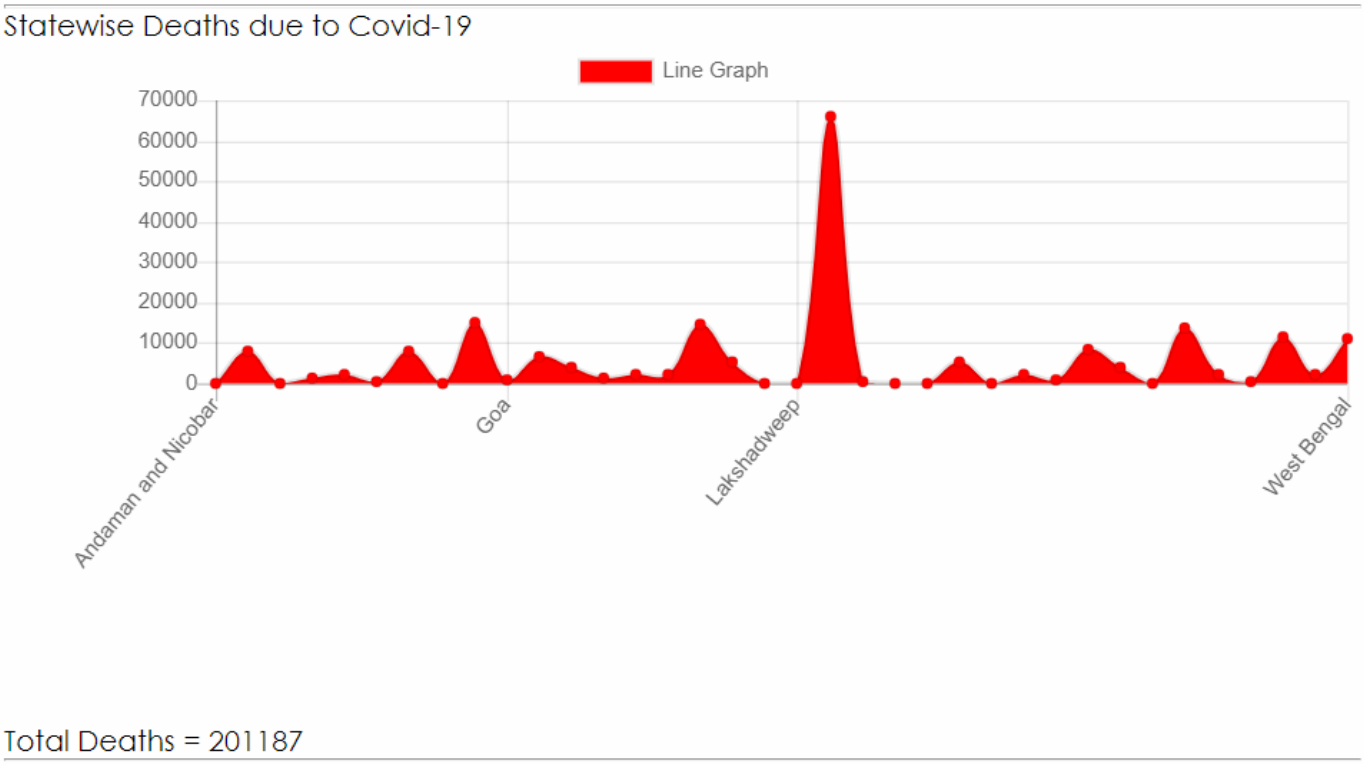


6.8 Total deaths state wise due to COVID-19 in India

Code:

```
var ctx_6 = document.getElementById("canvas5").getContext('2d');
var labels5=[];
var dataPoints5=[];
var total_deaths=0;
function addData6(data) {

for(var i=0;i<data.length;i++){
  labels5.push(data[i]["Name of State / UT"])
  // console.log("data32",data);
}

for(var j=0;j<data.length;j++){
  dataPoints5.push(data[j]["Deaths"])
  total_deaths +=parseInt(data[j]["Deaths"]);
}
document.getElementById("total_deaths").innerText = "Total Deaths = "+ total_deaths;
}
var chart = new Chart(ctx_6, {
    type: 'line',
    data: {
        labels: labels5,
       datasets: [{
            label: 'Line Graph',
            data: dataPoints5,
            backgroundColor: 'red',

       }]
    }
});
```

Graph:



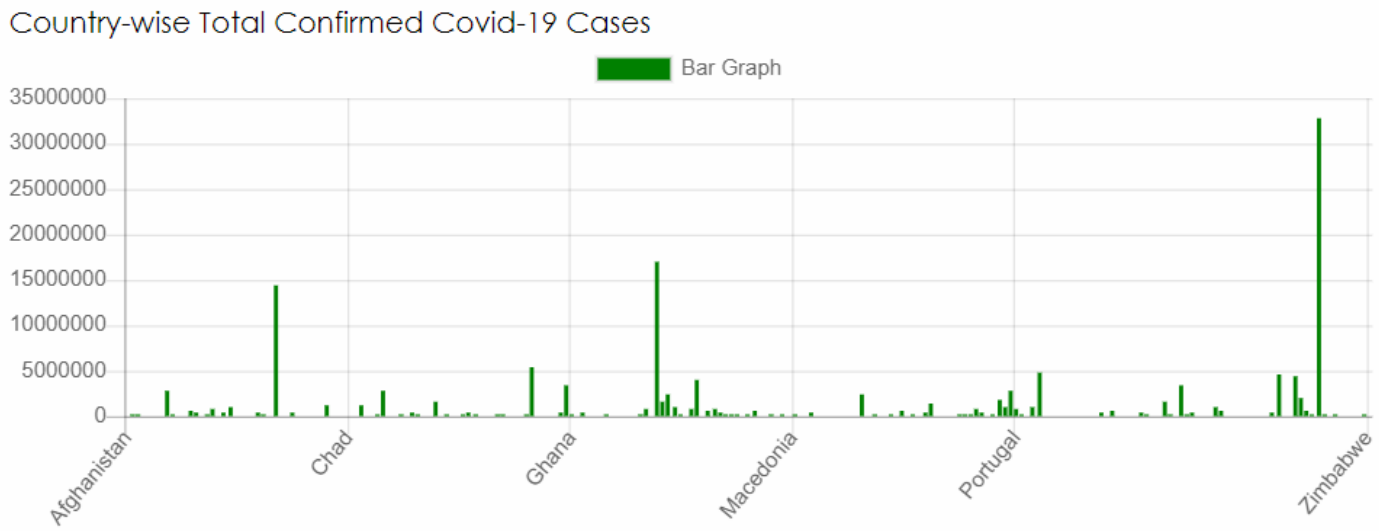
6.9 Total Confirmed COVID-19 cases country-wise

Code:

```
var ctx_3 = document.getElementById("canvas2").getContext('2d');
var labels2=[];
var dataPoints2=[];
function addData3(data) {

for(var i=0;i<data.length;i++){
  labels2.push(data[i]['country'])
  // console.log("data32",data);
}

for(var j=0;j<data.length;j++){
  dataPoints2.push(data[j]['total_confirmed'])
}
}
var chart = new Chart(ctx_3, {
    type: 'bar',
    data: {
        labels: labels2,
       datasets: [{
            label: 'Bar Graph',
            data: dataPoints2,
            backgroundColor: 'green',

       }]
    }
});
```

User has to click on **INTERNATIONAL STATS** button to view international statistics.

Graph:

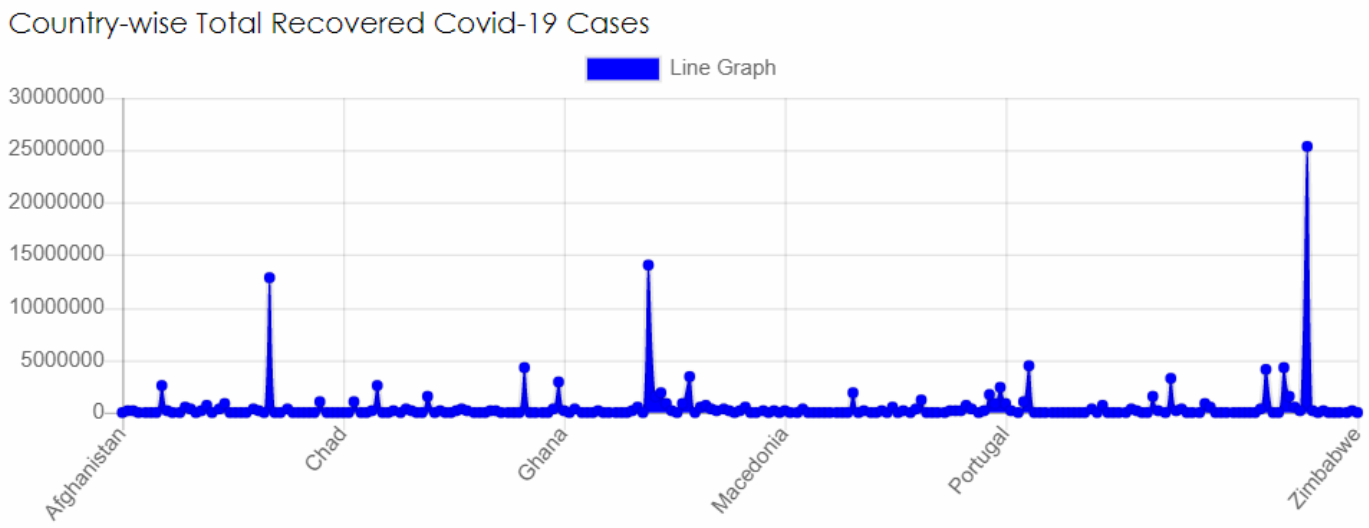


6.10 Total COVID-19 recovered cases country-wise

Code:

```
var ctx_4 = document.getElementById("canvas3").getContext('2d');
var labels3=[];
var dataPoints3=[];
function addData4(data) {

for(var i=0;i<data.length;i++){
  labels3.push(data[i]['country'])
  // console.log("data32",data);
}

for(var j=0;j<data.length;j++){
  dataPoints3.push(data[j]['total_recovered'])
}
}
var chart = new Chart(ctx_4, {
    type: 'line',
    data: {
        labels: labels3,
        datasets: [{
            label: 'Line Graph',
            data: dataPoints3,
            backgroundColor: 'blue',

        }]
    }
});
```

Graph:

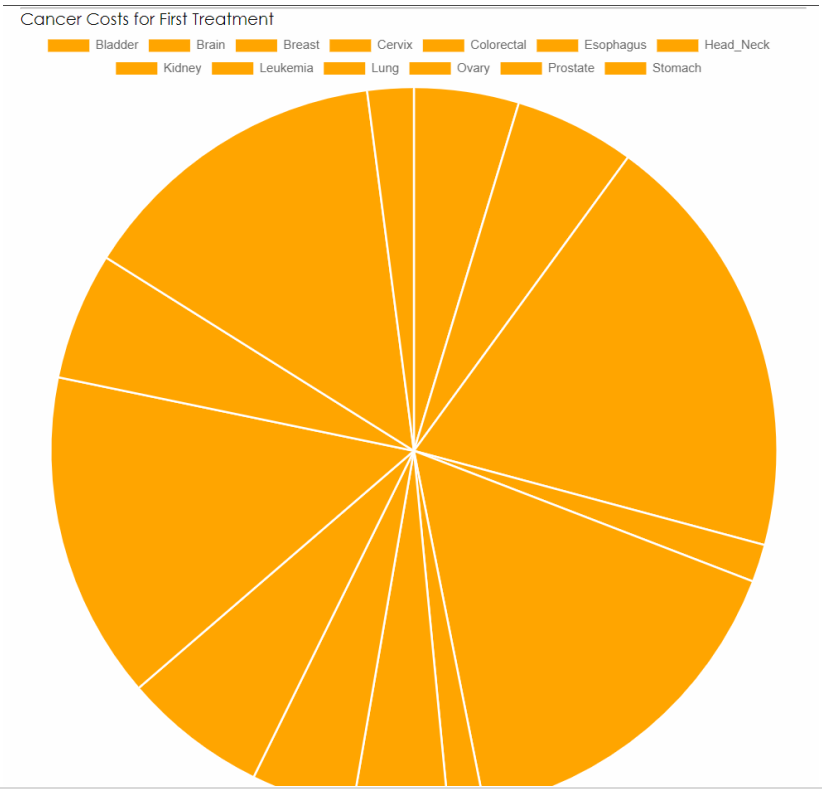


6.11 Graphical Analysis of Other Major Diseases like Cancer

Code (Cancer treatment costs):

```
var ctx_7 = document.getElementById("canvas6").getContext('2d');
var labels6=[];
var dataPoints6=[];
function addData7(data) {

for(var i=0;i<data.length;i++){
  labels6.push(data[i]["Cancer Site"])
  // console.log("data32",data);
}

for(var j=0;j<data.length;j++){
  dataPoints6.push(data[j]["Total Costs"])
}
}
var chart = new Chart(ctx_7, {
    type: 'pie',
    data: {
        labels: labels6,
       datasets: [{
            label: 'Pie Chart',
            data: dataPoints6,
            backgroundColor: 'orange',

       }]
    }
});
```

Graph:



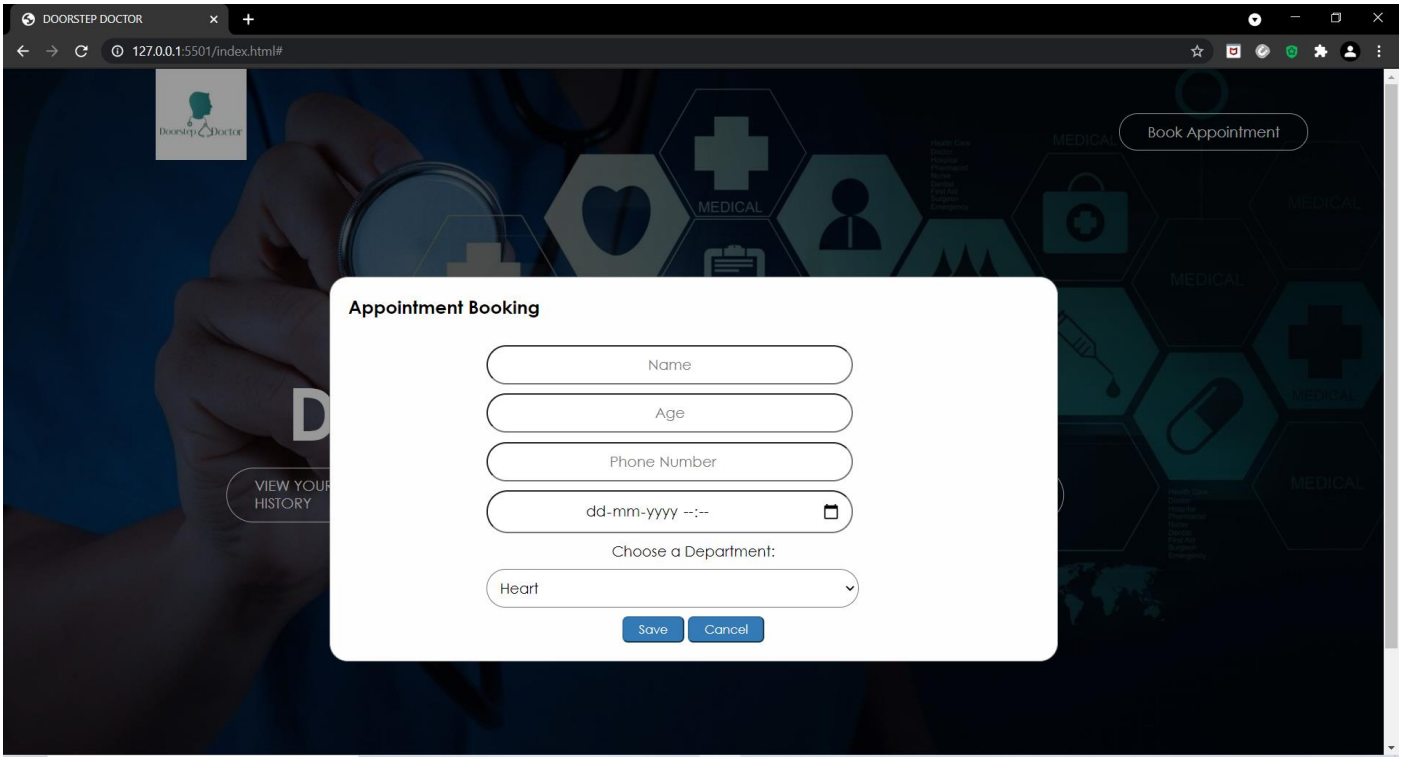
6.12 Patient Appointment Booking System

Code:

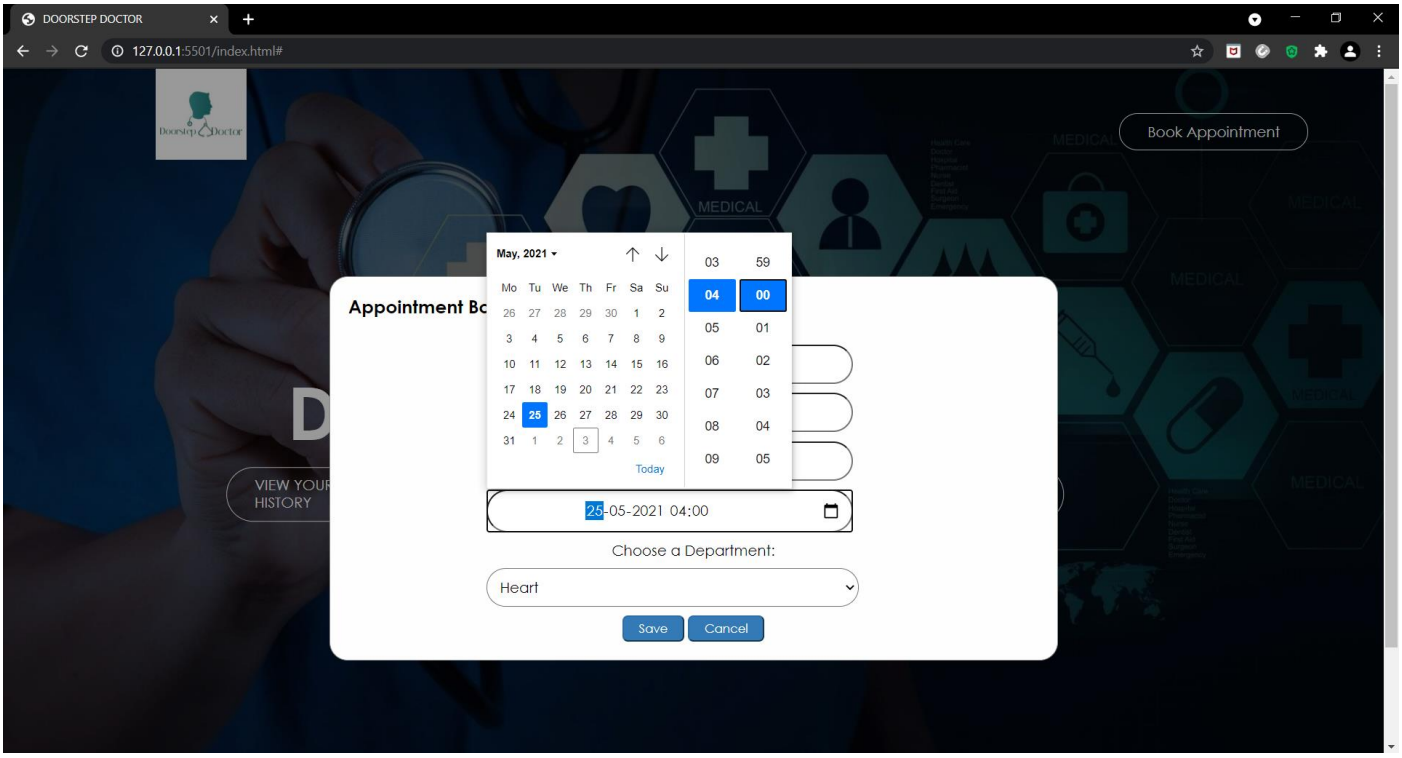
```
<div id="myModal4" class="modal">
  <!-- Modal content -->
  <div class="modal-content">
    <div class="modal-head">
      <p>Appointment Booking</p>
    </div>
    <br />
    <div class="modal-body">

      <form action="#" id="signupform" >
        <input placeholder="Name" class="popup__form">
        <input placeholder="Age" class="popup__form">
        <input type="number" placeholder="Phone Number" class="popup__form">
        <input type="datetime-local" placeholder="Date - Time" class="popup__form">
        <div class="ddown">
          <label for="cars">Choose a Department:</label>

<select name="depts" id="depts" class="popup__form" style="width: 54%;">
  <option value="volvo">Heart</option>
  <option value="saab">Renal</option>
  <option value="mercedes">Gastro</option>
  <option value="audi">Covid</option>
  <option value="audi">Physician</option>
</select>

        </div>
        <div style="text-align: center;">

          <button class="btnPrimary" id="saveBtn1"  onclick="myFunction()">Save</button>
          <button class="btnPrimary" id="cancelBtn1" >Cancel</button>
        </div>

      </form>
```

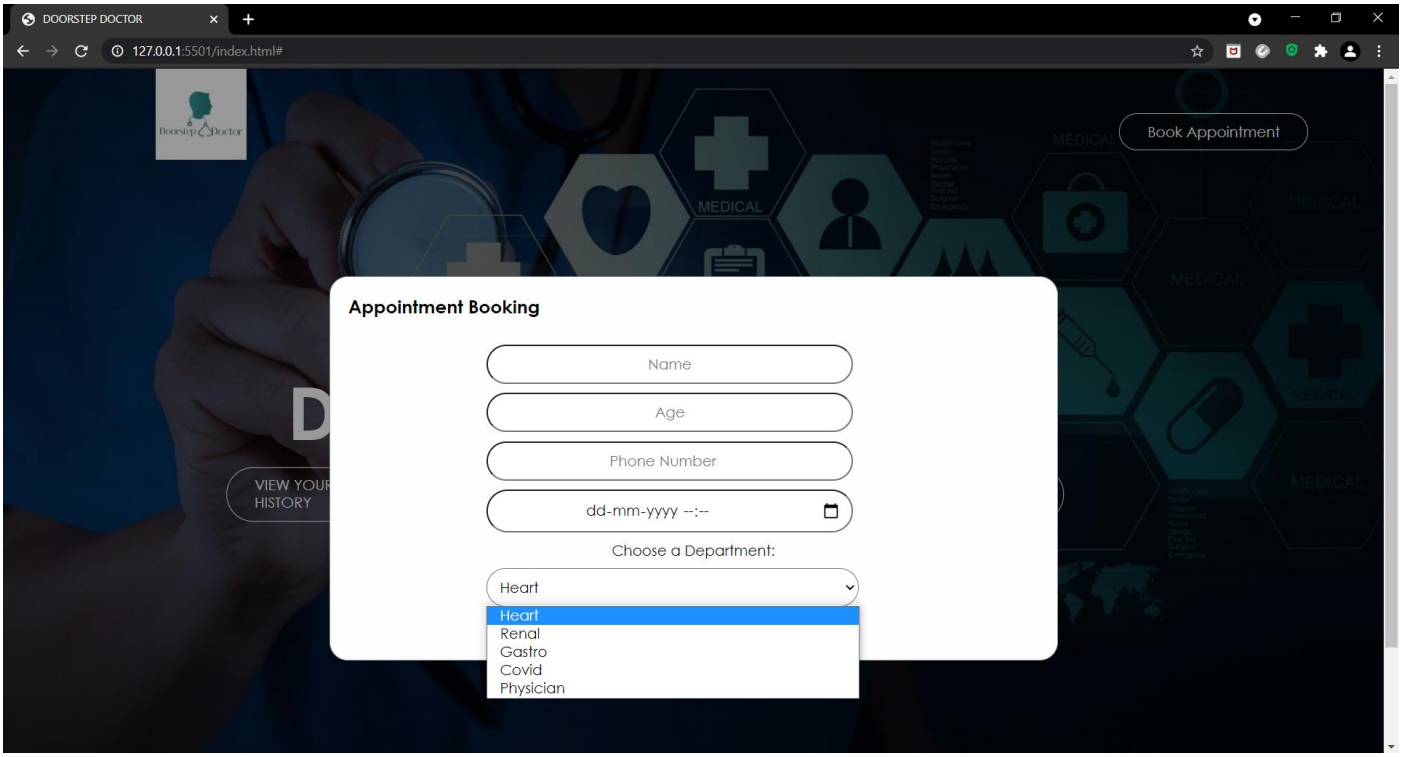
Patient need to click on **BOOK APPOINTMENT** button

On clicking a form will be opened for user to enter his/her details



Patient can fill his/her **personal details**, choose **date and time** and **department of the Doctor** with whom they want to consult





On clicking the **SAVE button** a message will be displayed regarding their appointment

7. Algorithms Used

I. Linear Regression:

Regression Analysis is a technique used to estimate the relationship between variables and predict the value of one variable (dependent variable) on the basis of other variables (independent variables).

Simple Linear Regression

• It depicts the relationship between a dependent variable and an independent variable.

• It considers one quantitative and independent variable X to predict the other quantitative, but dependent, variable Y.

• A straight line is fit to the data.

Multiple Linear Regression

• It predicts the value of a variable based on the value of two or more other variables.

• It considers more than one quantitative and qualitative variable ($X_1 \ldots X_n$) to predict a quantitative and dependent variable Y

$$y = k + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Dependent Variable

Intercept

Coefficient

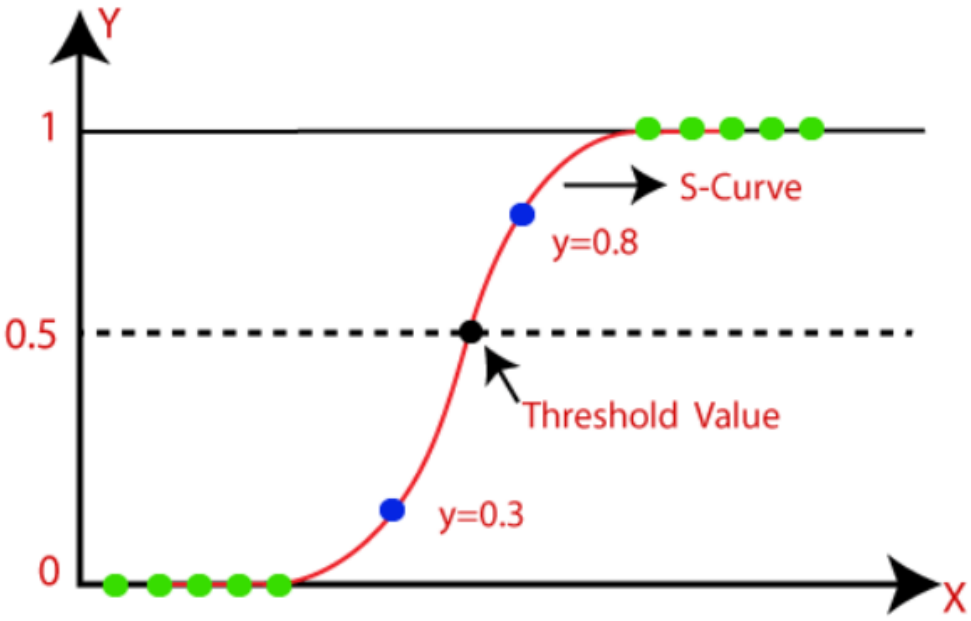
Predictors

II. Logistic Regression:

- Logistic regression is another technique borrowed by machine learning from the field of statistics.

- It is the go-to method for binary classification problems (problems with two class values).

- The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.



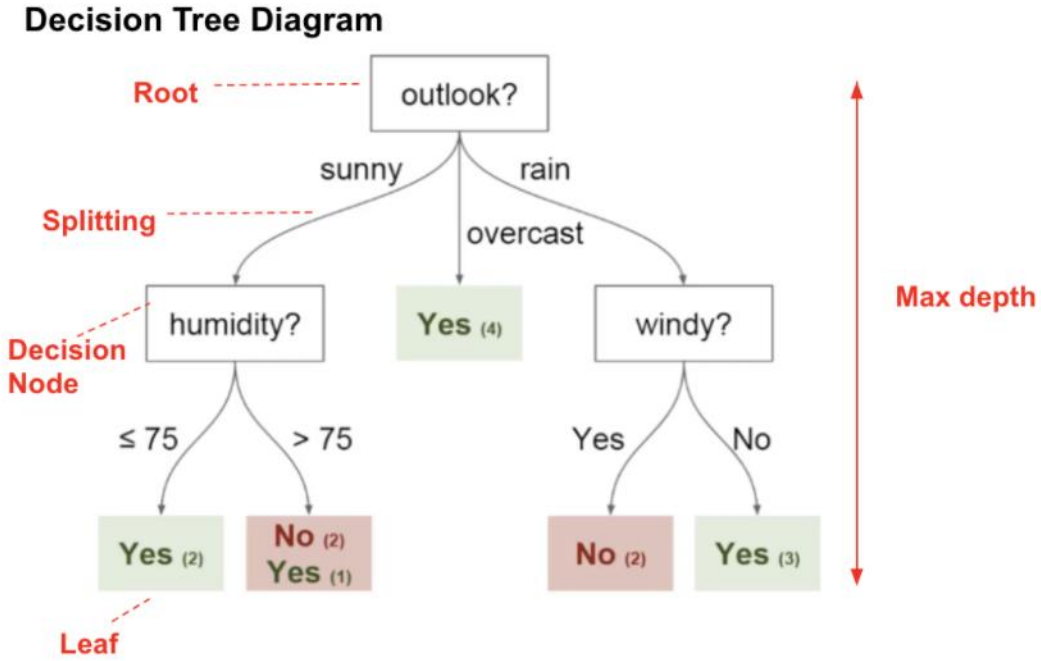
III. CART - Classification And Regression Trees:

- Classification And Regression Trees : Classification and Regression Trees or CART for short is a term introduced by Leo Breiman to refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems.

- Classically, this algorithm is referred to as "decision trees", but on some platforms like R they are referred to by the more modern term CART. The CART algorithm

provides a foundation for important algorithms like bagged decision trees, random forest and boosted decision trees. The representation for the CART model is a binary tree.

- This is your binary tree from algorithms and data structures, nothing too fancy. Each root node represents a single input variable (x) and a split point on that variable (assuming the variable is numeric). The leaf nodes of the tree contain an output variable (y) which is used to make a prediction.

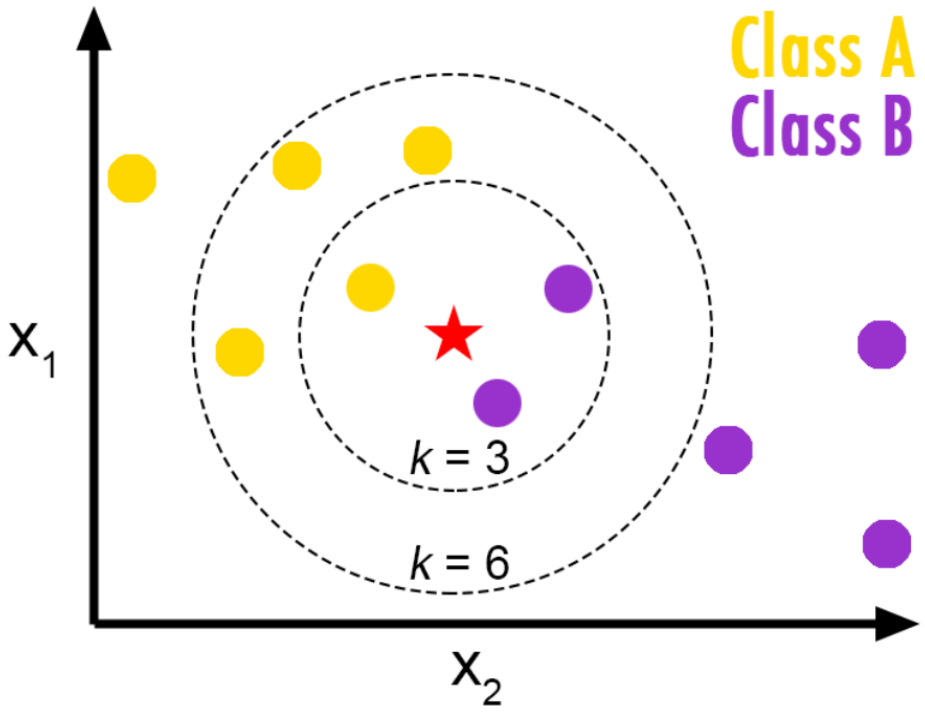


IV. KNN – K Nearest Neighbours:

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

The performance of the k-nearest neighbourhoods (k-NN) classifier is highly dependent on the distance metric used to identify the k nearest neighbours of the query points. The standard Euclidean distance is commonly used in practice.

The k-NN algorithm increases the accuracy of diagnosis. The algorithm can be used to enhance the automated diagnoses, which include diagnosis of multiple diseases showing similar symptoms.



8. Risk Analysis

Data visualization is the process of transforming text-heavy data into pictorial or graphical formats like infographics, dials and gauges, geographic maps, spark lines, heat maps, and detailed bar, pie and fever charts. Information comes to life with graphics, making relevant data difficult to overlook. However, if the same information is hidden in mind-blurring columns, rows, cells and text, valuable insights could be lost.

Even the most eye-catching dashboards, charts and graphs equate to “lipstick on a pig” if they reflect incomplete or out-of-date information; if they showcase surface-level or static data; or even if they are difficult to produce.

Another data challenge in healthcare relates to the decentralized nature of the industry. Major systems may be digital, but many organizations and smaller providers are still paper-based. AI and many other technologies can only operate using digital information. This fragmentation of datasets makes it difficult to develop a complete picture of healthcare scenarios.

Privacy restrictions imposed by federal law regulate the release of medical information. Data fragmentation and the lack of uniform digitization impede efficiency, with some data overlooked because it’s stuck in silos. If collecting data violates patients’ expectations for privacy, however, any gains are lost due to noncompliance. Furthermore, if data collection isn’t transparent, individuals are less likely to share the kind of information necessary for protecting society during the next pandemic.

9. Implementation

9.1 Analysis:

9.1.1- Front-end Implementation:

HTML and JavaScript:

index.html × # style.css

index.html > html > body > div#myModal2.modal > div.modal-content > div.modal-head > span.close

```
<!DOCTYPE html>
<html>
  <head>
    <title>DOORSTEP DOCTOR</title>
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.6.0/Chart.min.js"></script>
  </head>
  <script>
    $(document).ready(function () {
      var data;

      $.ajax({
        url: "http://localhost:3001/hospitalRecord",
        context: document.body,
        success: function (response) {
          // console.log("response", response);
          data = response;
        },
      });

      $("#bookAppointment").click(function(){
        $("#myModal4").show();
      })

      $("#medHistory").click(function () {
        $("#myModal").show();
    });

    $("#viewRecords").click(function () {
        $("#myModal2").show();
    });
```

CSS:

index.html # style.css

css > # style.css > .popup_form

```
*{
    margin: 0;
    padding: 0;
    font-family:   Century Gothic;
}

header{
    background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)), url(../back.jpg);
    height: 100vh;
    background-size: cover;
    background-position: center;
}

ul{
    float: right;
    list-style-type: none;
    margin-top: 25px;

}

ul li{
    display: inline-block;
}

ul li a{
    text-decoration:none;
    color: #fff;
    padding: 5px 20px;
    border: 1px solid transparent;
    transition: 0.6s ease;
}

```

9.1.2- Back-end Implementation:

Back-end is designed in JavaScript using NodeJS (Express Module):

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

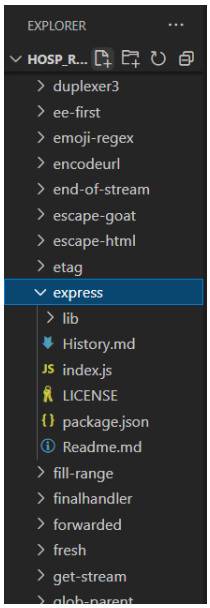
app.js package-lock.json

app.js > ...

```
const express = require('express')
const MongoClient = require("mongodb").MongoClient;

const cors=require('cors');


const CONNECTION_URL ='mongodb://localhost:27017/' ;
const DATABASE_NAME = "hospRecord";

const app = express()
const port = 3001

app.use(cors());

var database, collection;

app.get("/hospitalRecord", (request, response) => {
    collection.find({}).toArray((error, result) => {
        if(error) {
            return response.status(500).send(error);
        }
        response.send(result);
    });
});



app.listen(port, () => {
MongoClient.connect(CONNECTION_URL, { useNewUrlParser: true }, (error, client) => {
    if(error) {
        throw error;
    }
    database = client.db(DATABASE_NAME);
```

NodeJS modules (Express Module is used):

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications



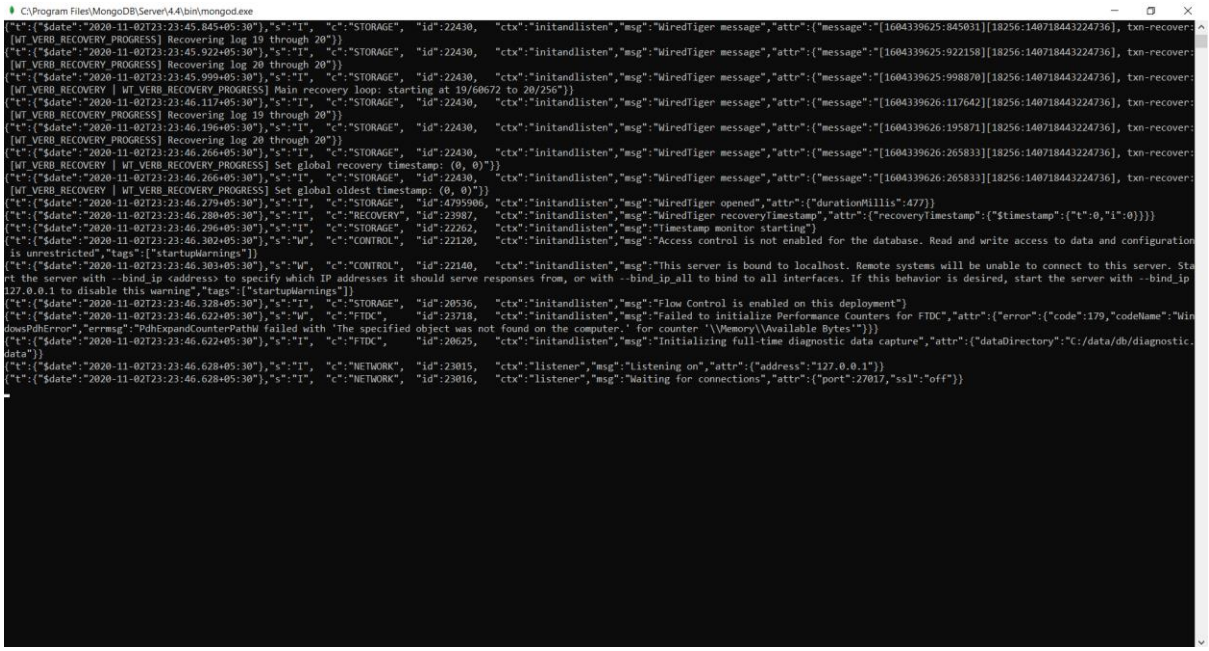
9.1.3- Database:

Database with name as “hospRecord” is created in MongoDB:

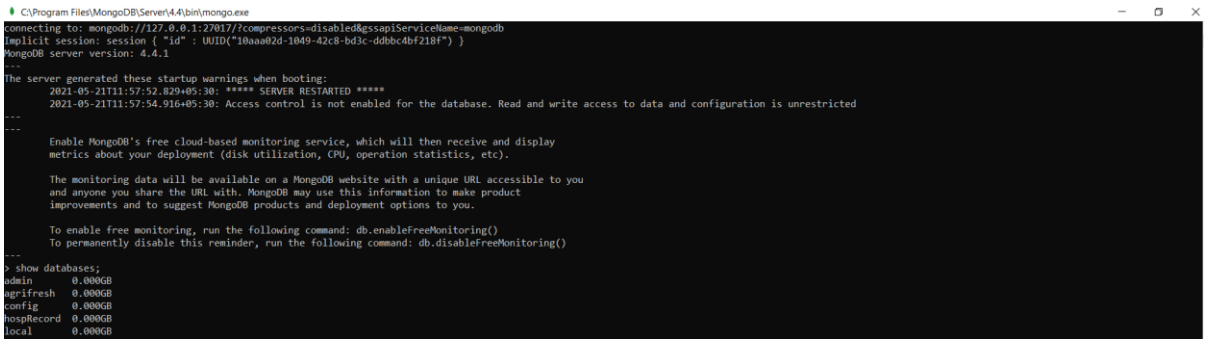
Information about MongoDB:

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

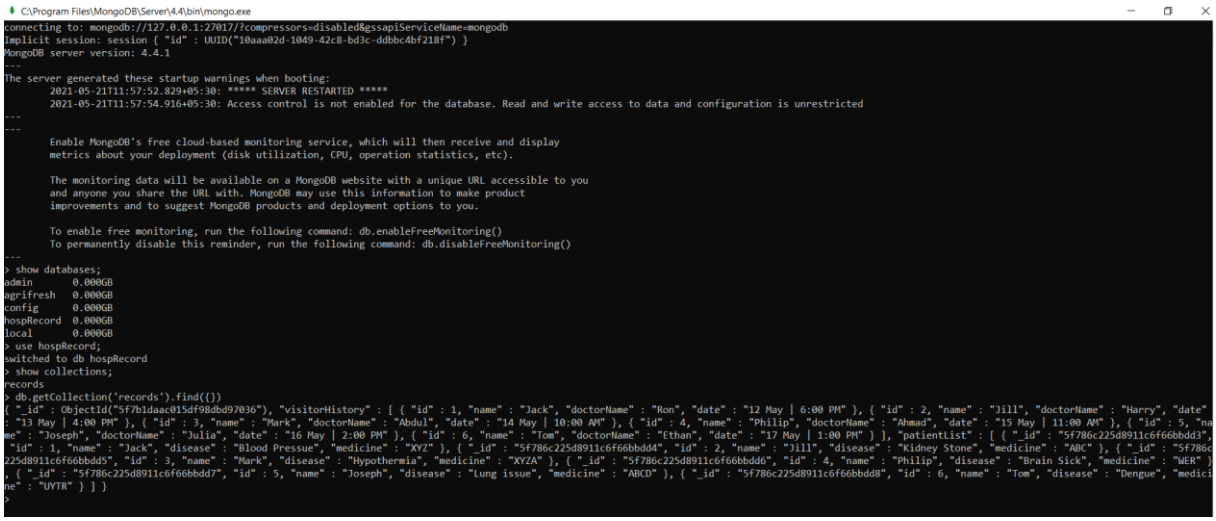
Mongo Server:



Showing "hospRecord" database:



Inside the database "hospRecord" a collection called "records" is created:



The collection "record" is made using JSON (JavaScript Object Notation):

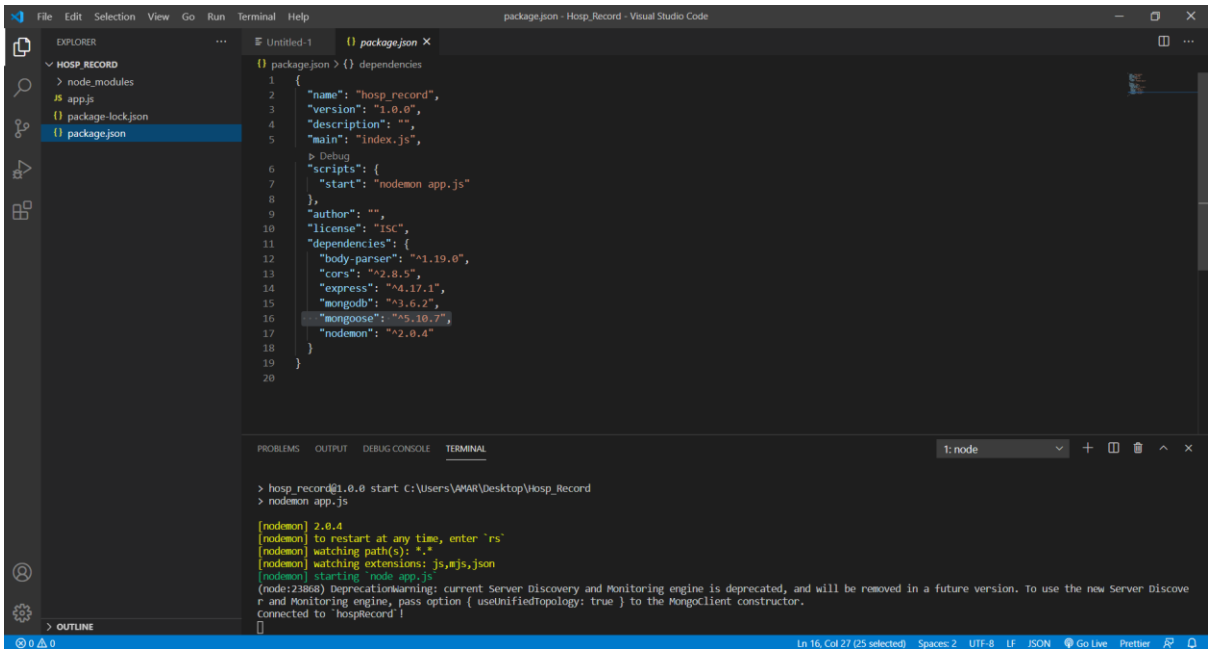
Information about JSON:

JavaScript Object Notation is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format, with a diverse range of applications, such as serving as a replacement for XML in AJAX systems.

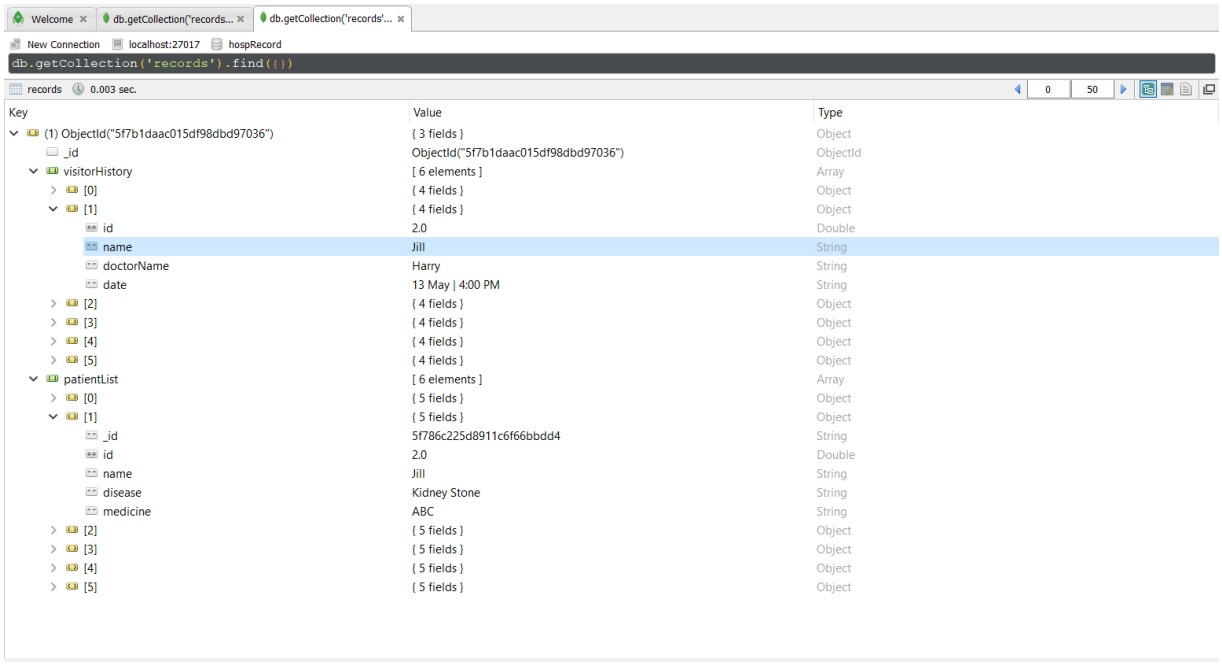
JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json. JSON filenames use the extension .json.

Douglas Crockford originally specified the JSON format in the early 2000s. JSON was first standardized in 2013, as ECMA-404. RFC 8259, published in 2017, is the current version of

the Internet Standard STD 90, and it remains consistent with ECMA-404. That same year, JSON was also standardized as ISO/IEC 21778:2017. The ECMA and ISO standards describe only the allowed syntax, whereas the RFC covers some security and interoperability considerations.



Showing **visitorHistory and patientList** inside hospRecords using Robo3T:



9.2 Visualization

9.2.1 Description

For visualization purpose we have used Chart.js and HTML5 Canvas.

HTML5 Canvas:

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolour rectangle, and a multicolour text.

What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

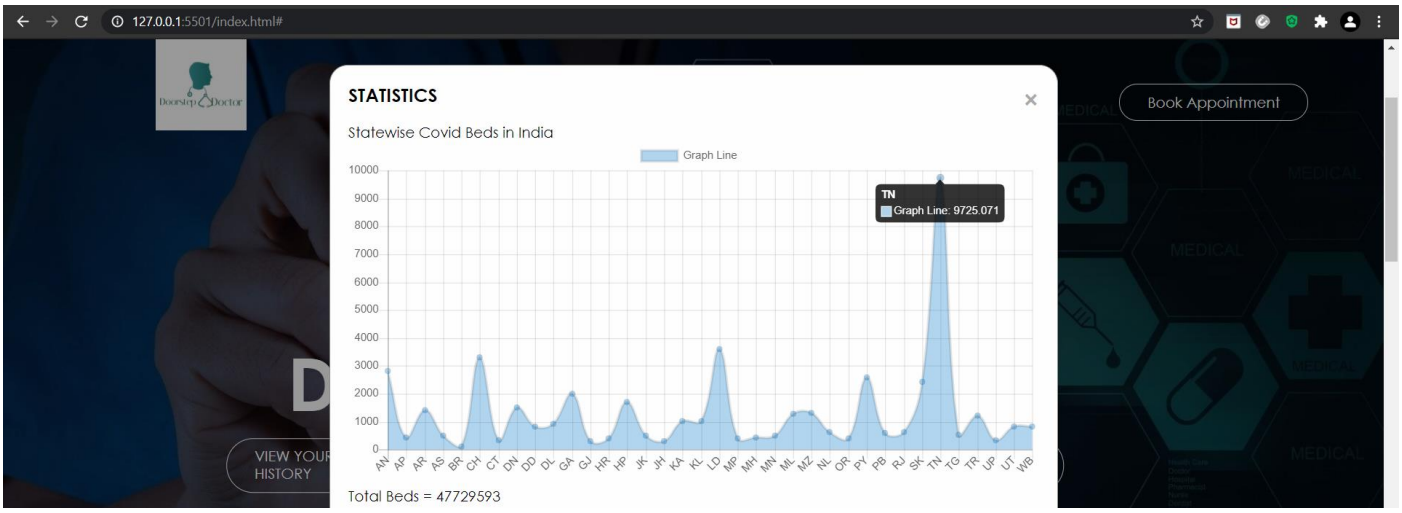
Chart.js:

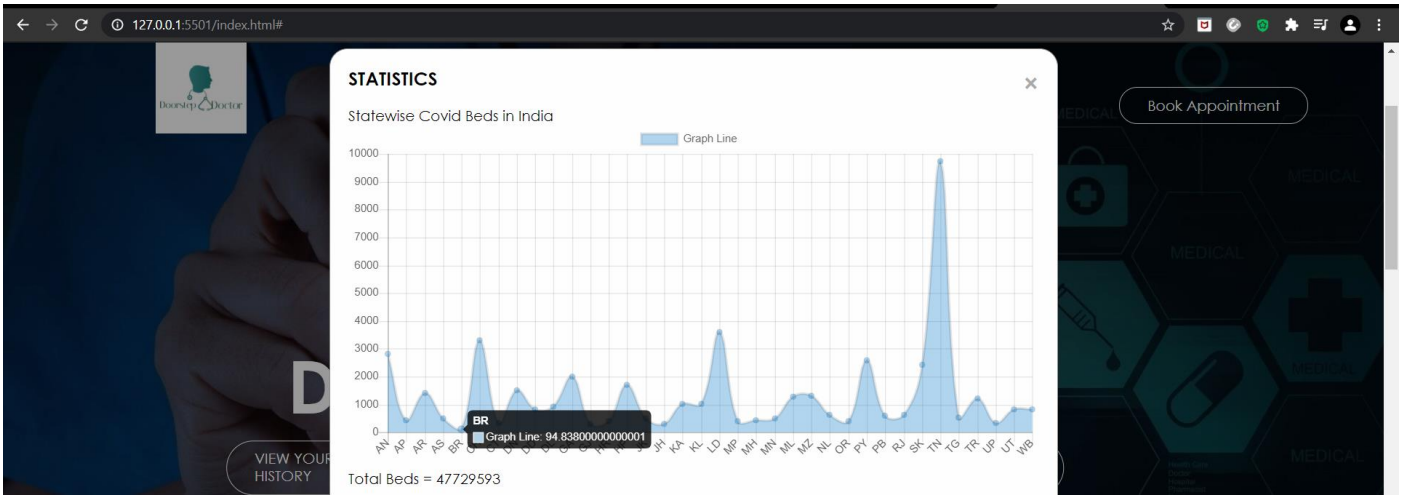
Chart.js is a free open-source JavaScript library for data visualization, which supports 8 chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. Created by London-based web developer Nick Downie in 2013, now it is maintained by the community and is the second most popular JS charting library on GitHub by the number of stars after D3.js, considered significantly easier to use though less customizable than the latter. Chart.js renders in HTML5 canvas and is widely covered as one of the best data visualization libraries. It is available under the MIT license.

9.2.2 Graphs

Under country stats we have plotted 5 graphs related to our Country data:

9.2.2.1 State-wise COVID-19 beds in India:

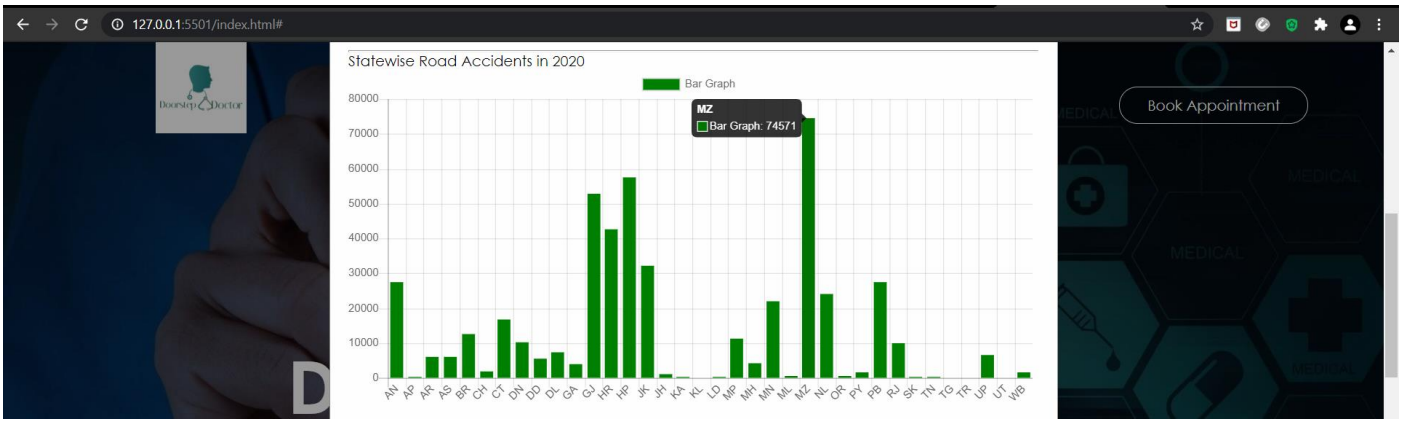


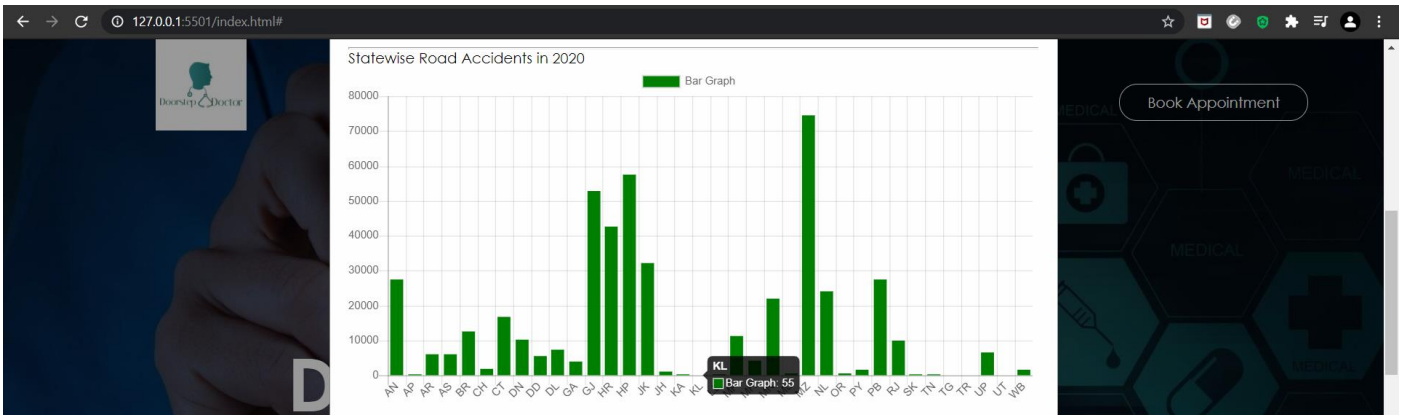


Conclusion from graphs:

- The graph shows total beds in the country.
- State/UT with highest number of beds is Tamil Nadu.
- State/UT with lowest number of beds is Bihar.

9.2.2.2 State-wise Road Accidents in the year 2020:

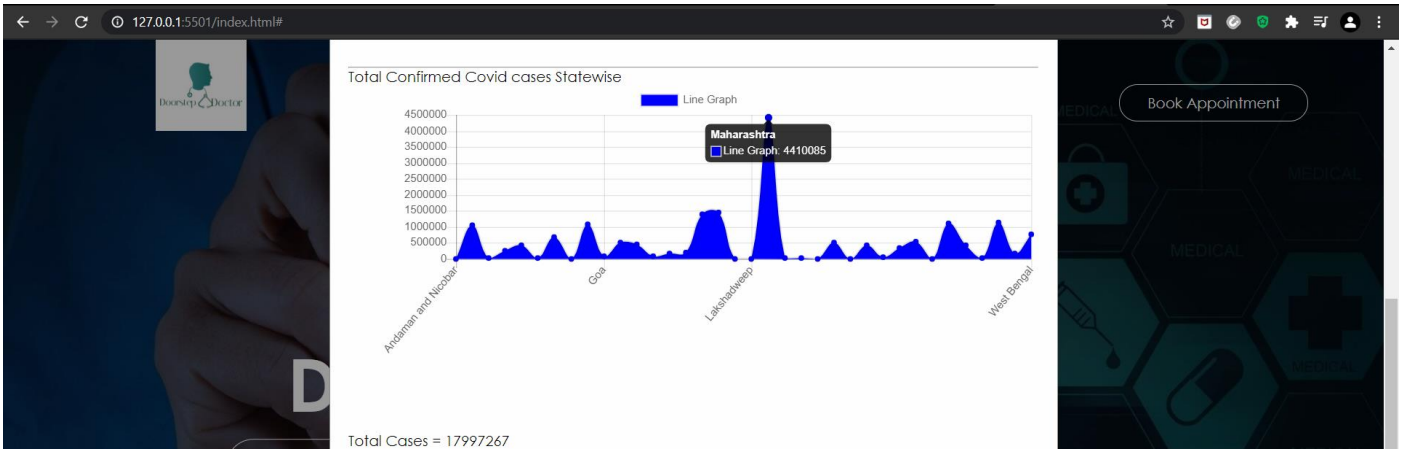


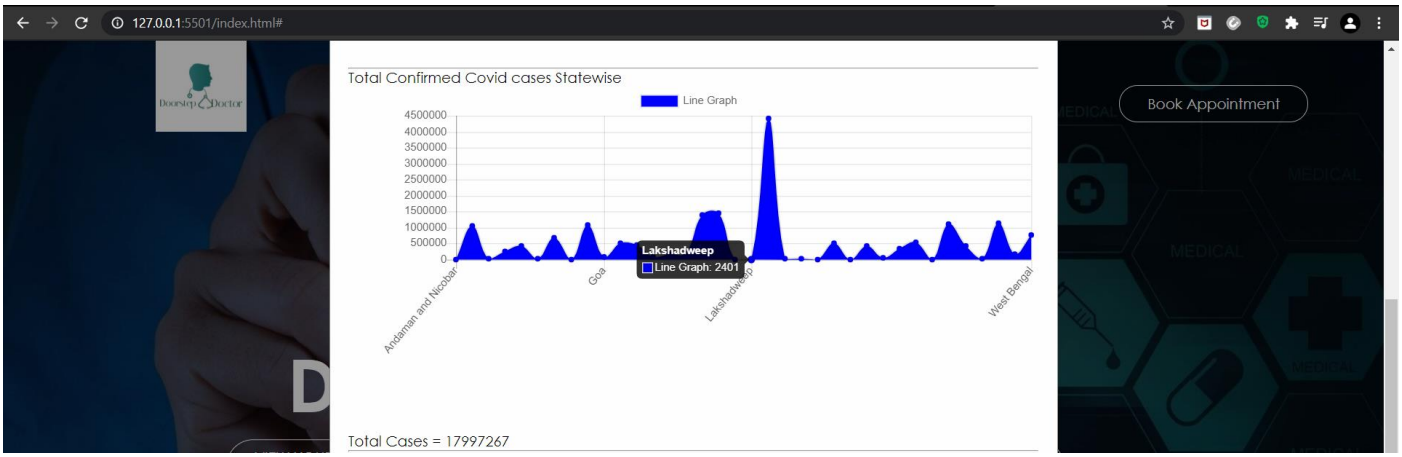


Conclusion from graphs:

- State/UT with highest number of road accidents in the year 2020 which is Mizoram.
- State/UT with lowest number of road accidents in the year 2020 which is Kerala.

9.2.2.3 Total confirmed COVID-19 cases state-wise:

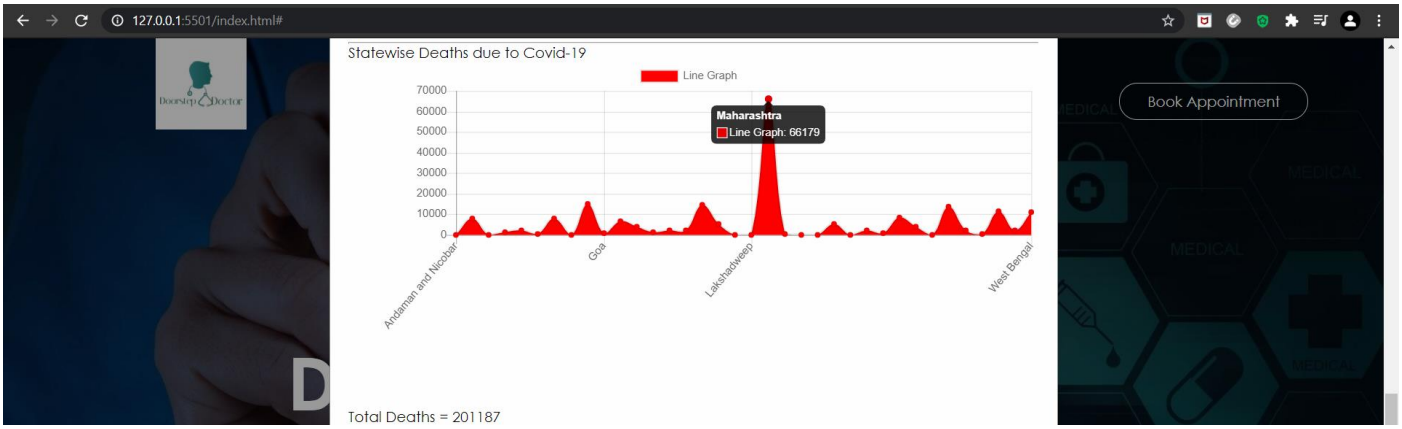


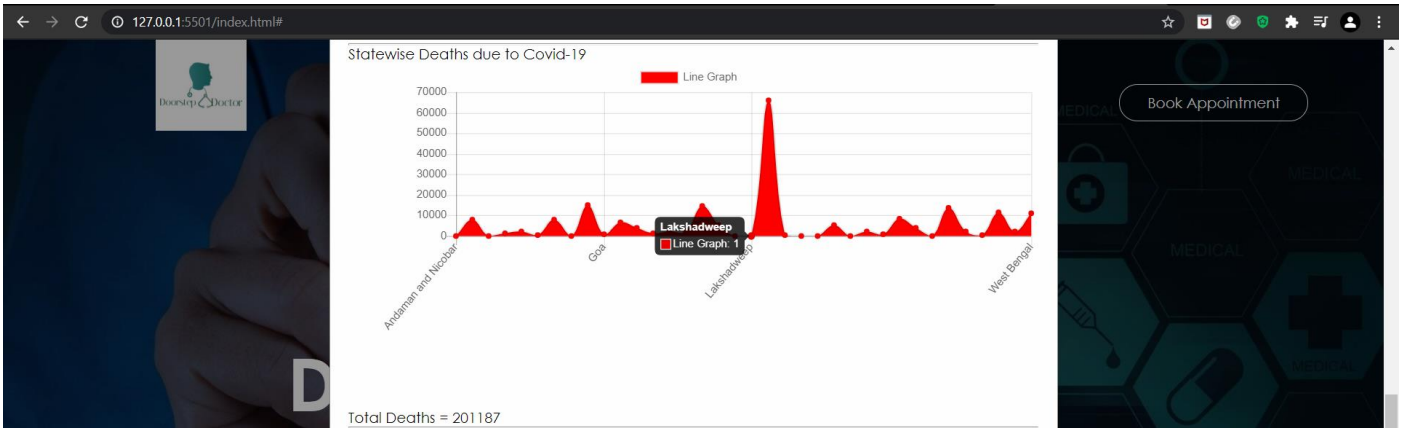


Conclusion from graphs:

- The graph shows total confirmed COVID-19 cases in the country.
- State/UT with highest number of COVID-19 confirmed cases is Maharashtra.
- State/UT with lowest number of COVID-19 confirmed cases is Lakshadweep.

9.2.2.4 State-wise deaths due to COVID-19

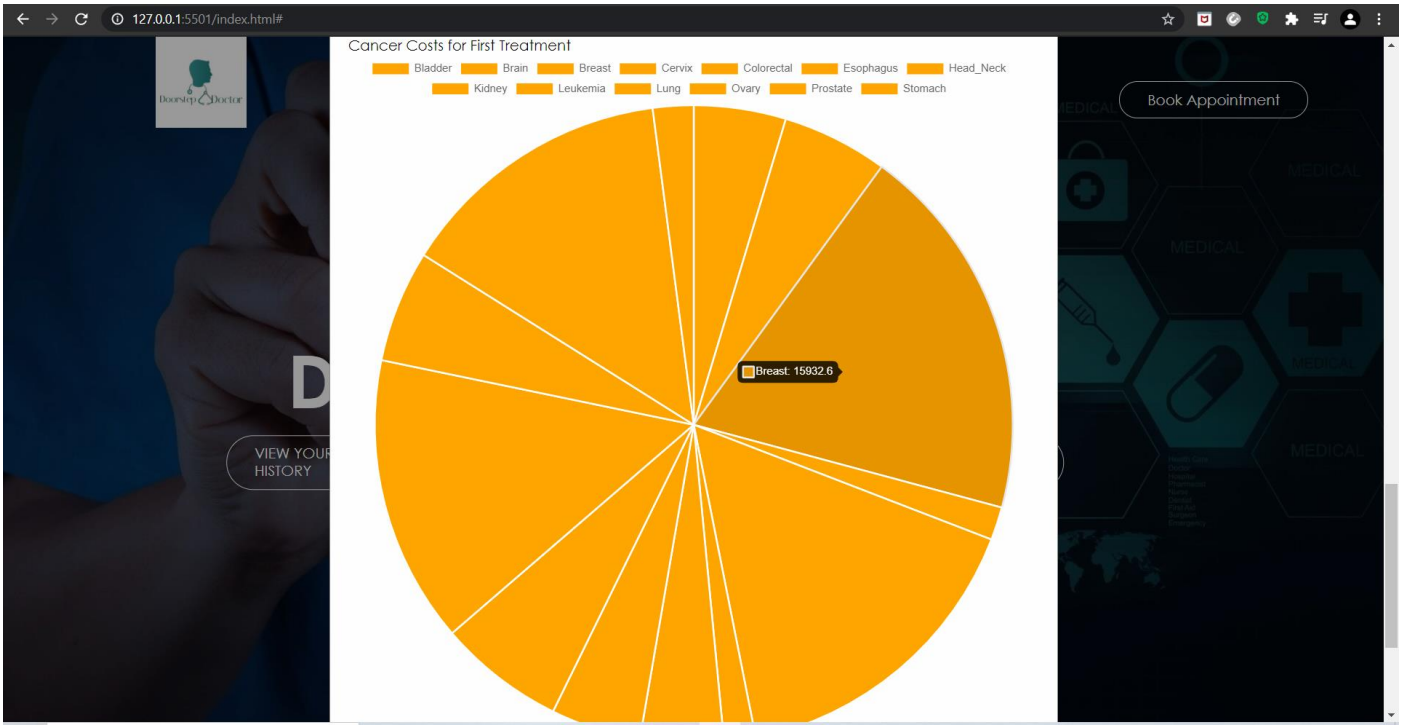


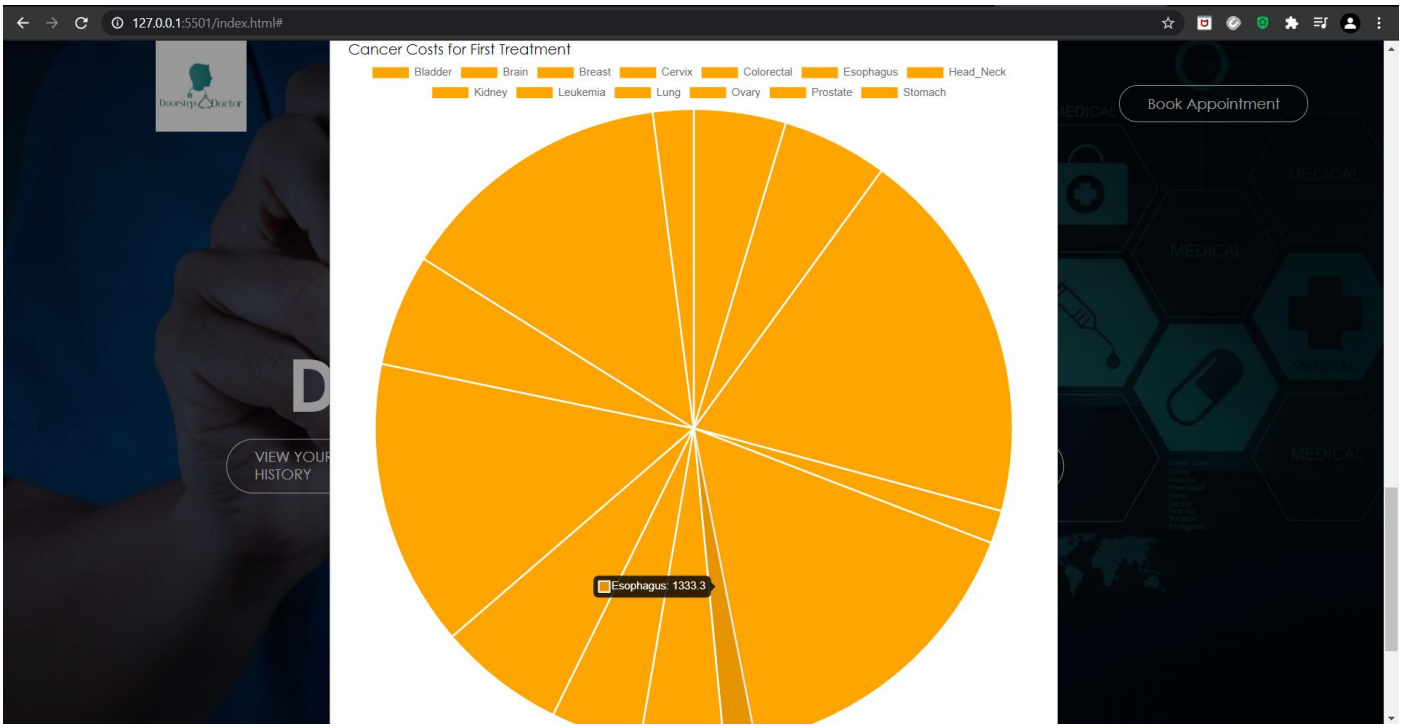


Conclusion from graphs:

- The graph shows total COVID-19 deaths in the country.
- State/UT with highest number of COVID-19 deaths is Maharashtra.
- State/UT with lowest number of COVID-19 deaths is Lakshadweep.

9.2.2.5 Cancer cost for First Treatment (in US $):



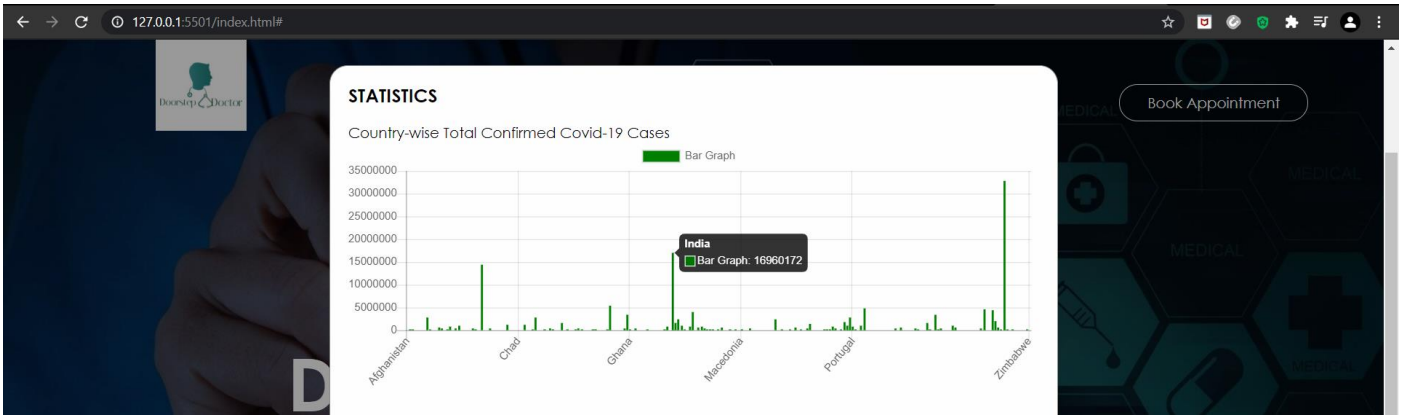


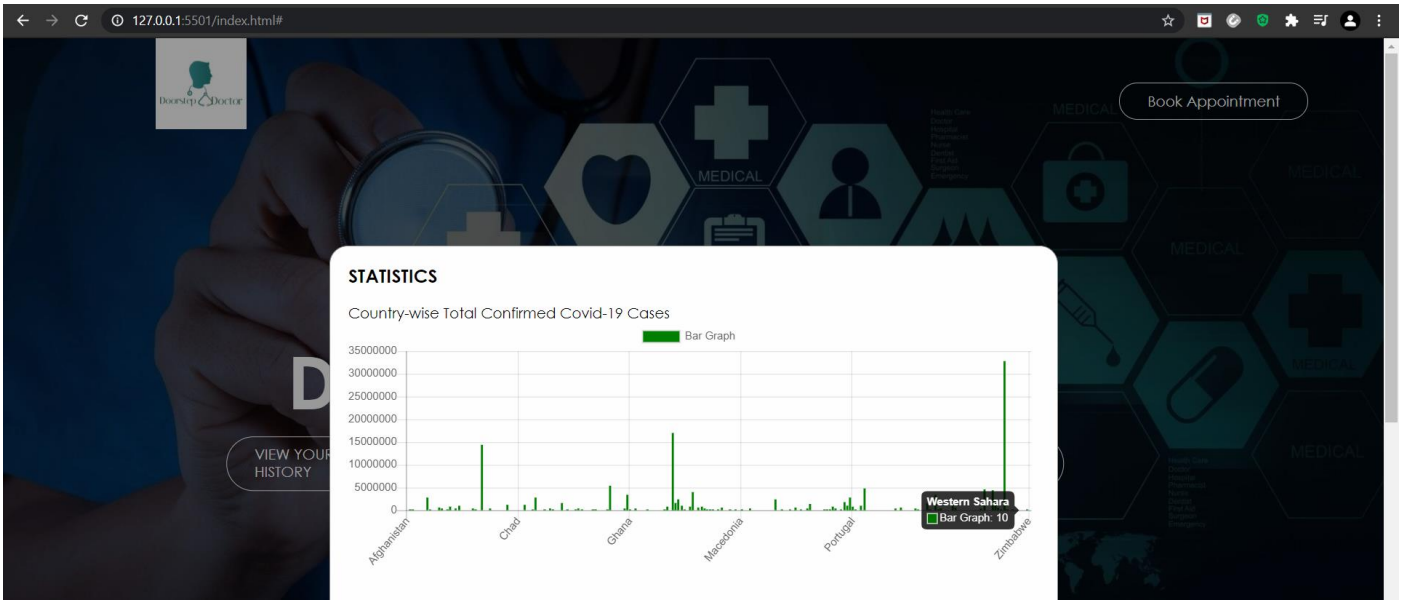
Conclusion from graphs:

- Cancer with highest cost for First Treatment (in US $) is Breast Cancer.
- Cancer with lowest cost for First Treatment (in US $) is Esophagus Cancer.

Under International stats we have plotted 2 graphs related to world-wide data:

9.2.2.6 Country-wise Total Confirmed Covid-19 Cases

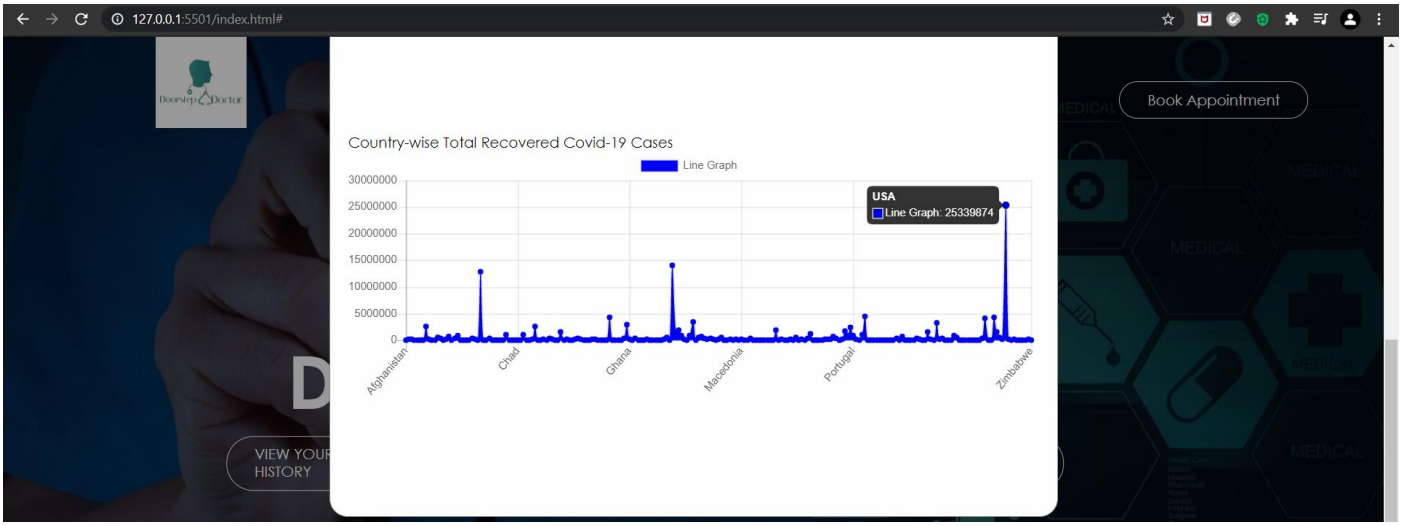


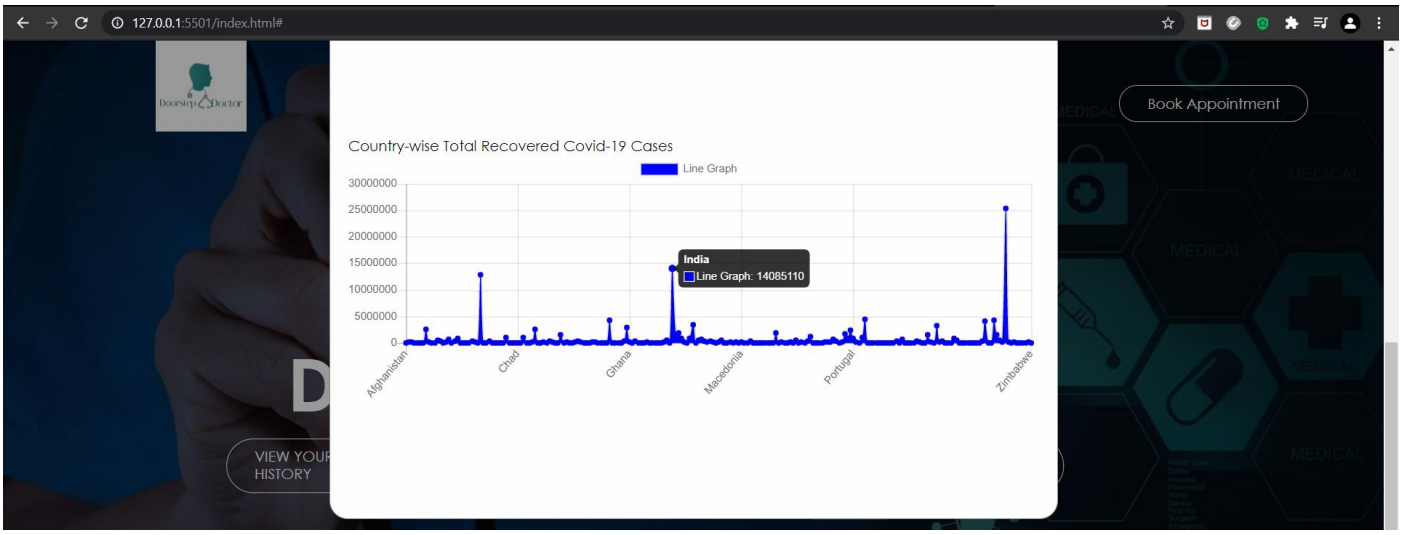


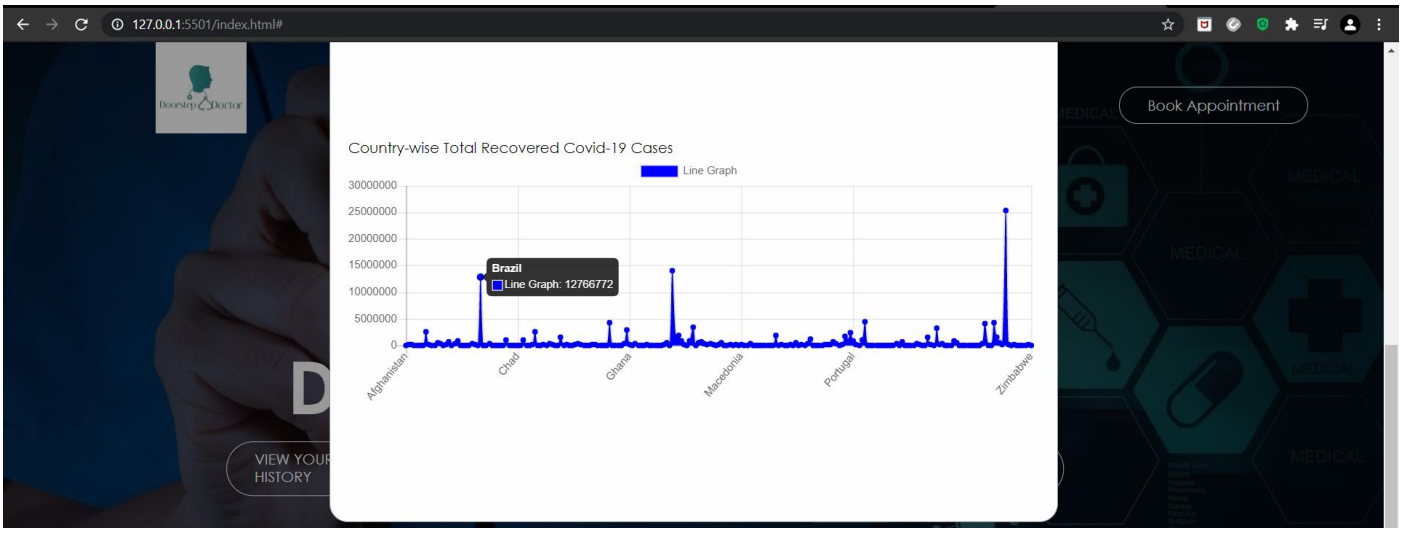
Conclusion from graphs:

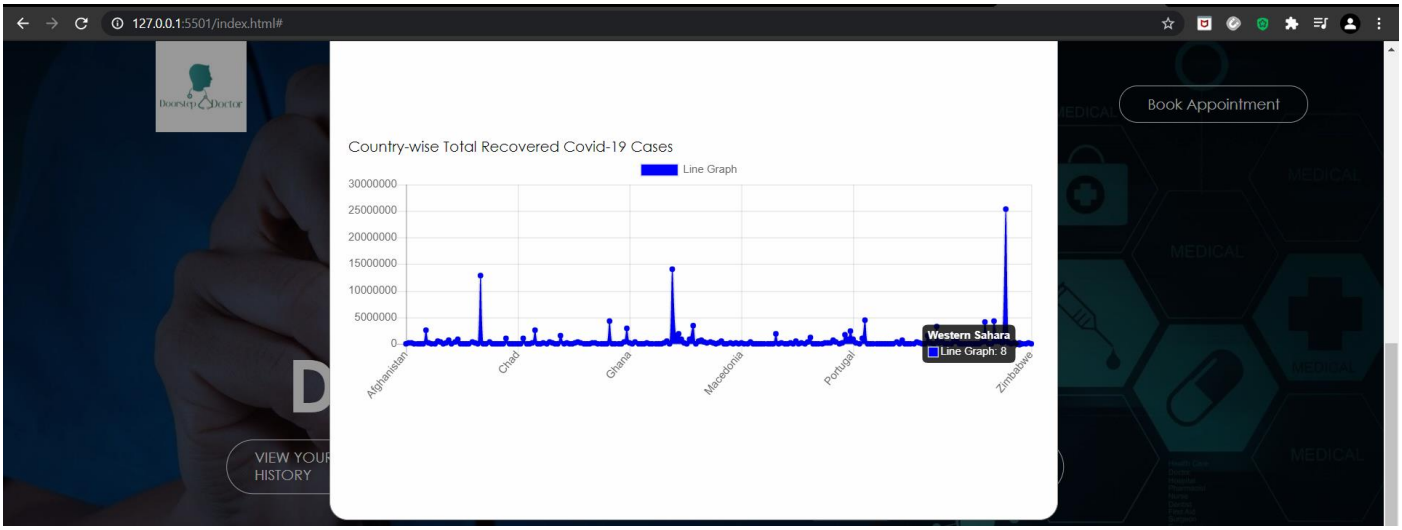
- Country with highest number of COVID-19 confirmed cases is USA.
- Country with 2nd highest number of COVID-19 confirmed cases is INDIA.
- Country with 3rd highest number of COVID-19 confirmed cases is BRAZIL.
- Region with lowest number of COVID-19 confirmed cases is WESTERN SAHARA.

9.2.2.7 Country-wise Total Recovered Covid-19 Cases:









Conclusion from graphs:

- Country with highest number of COVID-19 deaths is USA.
- Country with 2nd highest number of COVID-19 deaths is INDIA.
- Country with 3rd highest number of COVID-19 deaths is BRAZIL.
- Region with lowest number of COVID-19 deaths is WESTERN SAHARA.

10. Results

In this project we were able to develop various modules for hospital management such as patient appointment booking, patient medical history check and so on.

Apart from these we were also able to develop various statistical modules such as analysis of road accidents, bed availability in hospitals, various statistics related to COVID-19 and even cancer treatment costs.

The non-statistical modules were implemented using web and dataset tools such as HTML, CSS, JavaScript, NodeJS, MongoDB and many more.

All the statistical analysis were shown by generating and plotting graphs using some widely used algorithms such as linear regression, logical regression etc. and some visualization libraries such as ChartJS.

Through all these, we were able to develop a system which could analyse various datasets related to healthcare and provide us with real time, accurate and easier to interpret plots and graphs along with the patient management system which could be of great use to the society in the future in healthcare sector.

11. Conclusions

In this project we have comprehensively studied and analysed data from various trusted and reliable sources. We used some popular algorithms to achieve our statistical analysis the form of graphs and charts.

The graphs were one of the most important results of our project as they clearly depicted the current situation of the world in various fields of the healthcare sector.

Amid this COVID-19 pandemic, we could clearly see the problems arising exponentially in the healthcare sector. Using data analysis, it becomes easier to interpret, understand and improve on the current situation because such plots and charts and easier as well as comfortable to be studied not only by the respective officials but also by the common people.

Very less portion of the healthcare sector has adopted data visualization into their account to work on. It can prove to be very beneficial in the coming future. A lot of work still remains to be done to completely include data analysis and visualization in the healthcare sector, but we are sure that once it is done it will make the healthcare sector reach new heights given its great techniques with better accuracies which have many great applications.

Further studies are required in order to understand why some methods work better or worse than others depending on the datasets and the sectors they are applied to. New techniques with better accuracy and performance trade-offs are expected to be proposed in the forthcoming future.

12. References

- https://www.mohfw.gov.in/
- https://www.kaggle.com/
- https://data.world/datasets/open-data
- https://www.bbc.com/news/world-asia-india-56891016
- https://en.wikipedia.org/wiki/Data_visualization
- https://www.chartjs.org/
- https://www.youtube.com/
- https://www.w3schools.com/
- https://www.tutorialspoint.com/index.htm
- https://javascript.info/