KATHMANDU UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Dhulikhel, Kavre

A Project Report on

COVID Mapping

[Code No: COMP 207]

(For the partial fulfillment of 2nd year/2nd Semester in Computer Science)

Submitted by:

Bigen Aryal (77)

Bikramaditya Prasad Subedi (75)

Milan Dhamala (73)

Rajeshwor Niroula (71)

Usta Adhikari (68)

Submitted to:

Department of Computer Science and Engineering

Submission Date: 28th September

Bona fide Certificate

This project work on

“COVID Mapping”

is the bonafide work of

Bigen Aryal (77)

Bikramaditya Prasad Subedi (75)

Milan Dhamala (73)

Rajeshwor Niroula (71)

Usta Adhikari (68)

who carried out the work under my supervision.

Project Supervisor

_____

Name: Manish Joshi

Academic Designation: Lecturer, DOCSE

Project Examiner

_____

Abstract

Given the circumstances of our country dealing with COVID, keeping track of the ongoing cases is a matter of life and death to the community or the whole country. Thus, staying updated on the go might help prevent possibly from reaching places that has been known to have been infected so a idea came of building a web app that basically keeps track of the COVID cases in a provincial manner. A choropleth map with a legend of all provinces shows the active cases, recovered cases and deaths on screen where the web map serves datas from authentic sources allowing user to be visually informed of the infected zones which will help take precautions before planning a short trip or even go for a grocery. Python is used to build the backend and handle the source code. Meanwhile, folium is used to visualize data that has been manipulated in Python using an interactive leaflet map enabling both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as a marker on the map. Since, the datas are for a provincial view opting in for manual data entry seemed to be better decision. As for the database, MySQL features are enough for our simple database plan.

Keywords: Django, Mapping, Choropleth

TABLE OF CONTENTS

Title **Page No.**

List of Figures

Figure **Page No.**

List of Table

Table **Page No.**

Acronyms

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

SQL: Structured Query Language

PC: Personal Computer

DSF: Django Software Foundation

WHO: World Health Organization

Chapter 1: Introduction

Project COVID Mapping is a web map used for keeping track of ongoing cases of COVID-19 providing an eagle's eye view of the ongoing cases to the general public. Our project displays COVID-19 cases i.e. (active, recovered and death) on a choropleth map from a database where the database is fed with datas from official sources via manual entry. A simple UI will allow users to switch between the active, recovered or death view on an Open Street map.

1.1. Background

The COVID-19 pandemic continues to take toll across the world, costing lives and bringing upheaval and change to societies and economics around the globe. Across the world, governments and health authorities are working together to find solutions to the COVID-19 pandemic, to protect people and get society back up and running. This pandemic has triggered an unprecedented demand for digital health technology solutions and has revealed successful solutions such as for population screening, tracking the infection, prioritizing the use and allocation of resources, and designing targeted responses. However, an uncomplicated and straightforward manifesto is missing for the population to get error-free facts about count of COVID infected. So, we also as a responsible student, want to contribute to society to fight against this global pandemic by building an uncomplicated and simple web map with legend showing us the infection rates on our screen from authentic sources giving precise data count of infected to be updated of condition around us.

1.2. Objectives

Our project COVID Mapping has the following objectives:

- To make effective response using the available data to the harshly hit centers
- To clearly indicate on a choropleth map as patterns, trends would be missed in a textual data
- To provide precise and accurate COVID data in provincial manner
- To decrease reliability over API's as we gather datas from official sources manually as soon as it is released

1.3 Motivation and Significance

Health professionals have long considered conventional mapping, and more recently geographic information systems (GIS), as critical tools in tracking and combating contagion. The maps as a communication tool has been in service of understanding and tracking infectious diseases, such as yellow fever, cholera and the 1918 influenza pandemic. Since then we have seen a revolution in applied health geography through mapping. While there is no one way to spatially plot the diffusion and effects of COVID-19, a few maps stand out. One of the most authoritative is overseen by the World Health Organization (WHO), which provides a country-by-country update of cases. Some firms like Carto, ESRI and MapBox have developed sophisticated software and accessible dashboards to help governments, businesses, health agencies and grassroots groups better respond and recover from COVID-19. Despite news reports and these firms have been valuable inputs for public health surveillance, staying abreast of current disease outbreaks requires scouring a continually growing number of disparate news sources and alert-services, resulting in information overload. Our motivation is to address this challenge through COVID Mapping, a web application for querying and visualizing unstructured reports on the current scenario.

During a pandemic, more than in any other public health situation, information system plays a critical role in managing data and other information at the speed the situation requires. They provide

essential evidence for taking actions, making the most informed decisions possible. Since COVID-19 can be transmitted through close proximity to affected individuals, it is very important to know the red zones of corona virus, and to avoid reaching these zones. To minimize the spread individuals are expected to self-isolate and stay away from the infected areas, and to do so we should have idea about infected zones. And our project COVID-Mapping is made with the same intention of providing insights to using visual representation.

Chapter 2: Related Works

The most useful maps and data visualizations of the COVID-19 pandemic has helped us understand, respond to and recover from the ongoing crisis. Several maps seek to estimate the potential spread of COVID-19 using a combination of computational methods, while others track the socio-economic impact of the pandemic. A new data-driven tool to assist city-level governments and health providers, especially in lower and middle-income settings, is in development to help plan and manage future infectious disease outbreaks and few other similar efforts have been briefed below:

Coronavirus COVID-19 Global Cases is a dashboard developed by a professor at Johns Hopkins University (JHU, 2020). It was developed to provide researchers, public health authorities, and the general public with a user-friendly tool to track the outbreak as it unfolds. The map is maintained in near real time throughout the day through a combination of manual and automated updating.

Fig-2.1

HealthMap is a freely accessible, automated electronic information system for monitoring, organizing, and visualizing reports of global disease outbreaks according to geography, time, and infectious disease agent. HealthMap acquires data from a variety of freely available electronic media sources (e.g. Eurosurveillance, Wildlife Disease Information Node) to obtain a comprehensive view of the current global state of infectious diseases (Boston Children's Hospital, 2020).

Fig-2.2

ECDC also launched a COVID-19 situation dashboard with interactive maps, graphs, tables allowing users to easily view, compare and export data on subnational levels of transmission in the EU/EEA and the UK, as well as enhanced data on subsets of cases, including age, gender, hospitalization and admission to intensive care (NITAG, n.d.). The situation dashboard is updated every day to visualize and disseminate data.

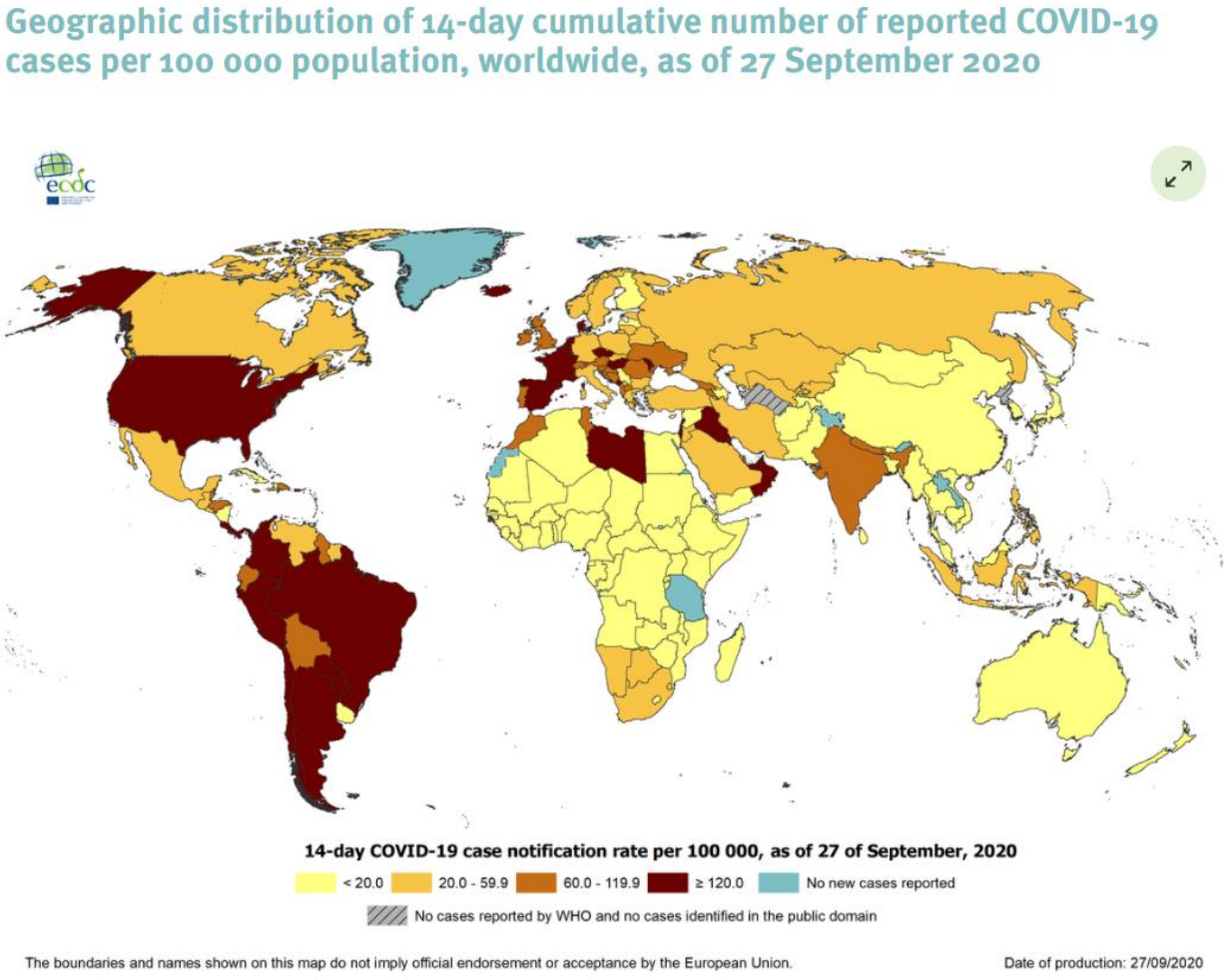

Fig-2.3

Worldometer is a reference website providing counters and real-time COVID-19 statistics to a wide audience around the world (Alimetov, n.d.). The data is collected from official reports, local medias and other channels directly or indirectly and is made available for viewing only when deemed reliable.

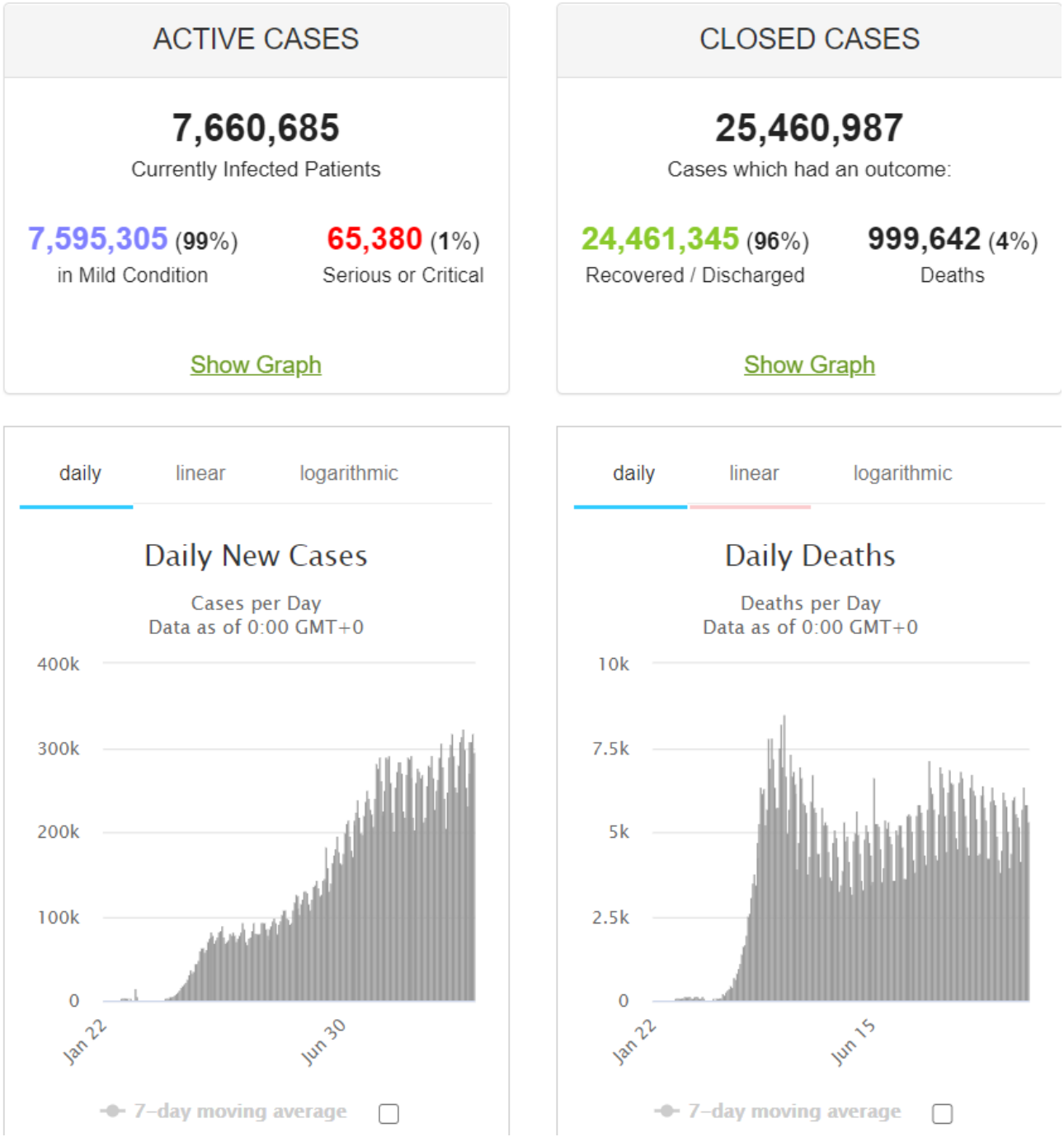

Fig-2.4

Chapter 3: Design and Implementation

Our project begins with installation of XAMPP for MySQL database and PyCharm IDE for Django development environment using a database model was built in Python that created database tables in MySQL database connected via Apache server as shown below in the flowchart. An app controller links the database model to all of our algorithm. App controller also takes data from the browser and feeds it to the controller which in turn updates the database model and tables. The updated models are converted to pandas dictionaries and are used by folium and Plotly to create interactive maps and bar plot which is passed to the controller as objects which is again forwarded to the view pages where it is rendered to HTML and displayed on the browser.

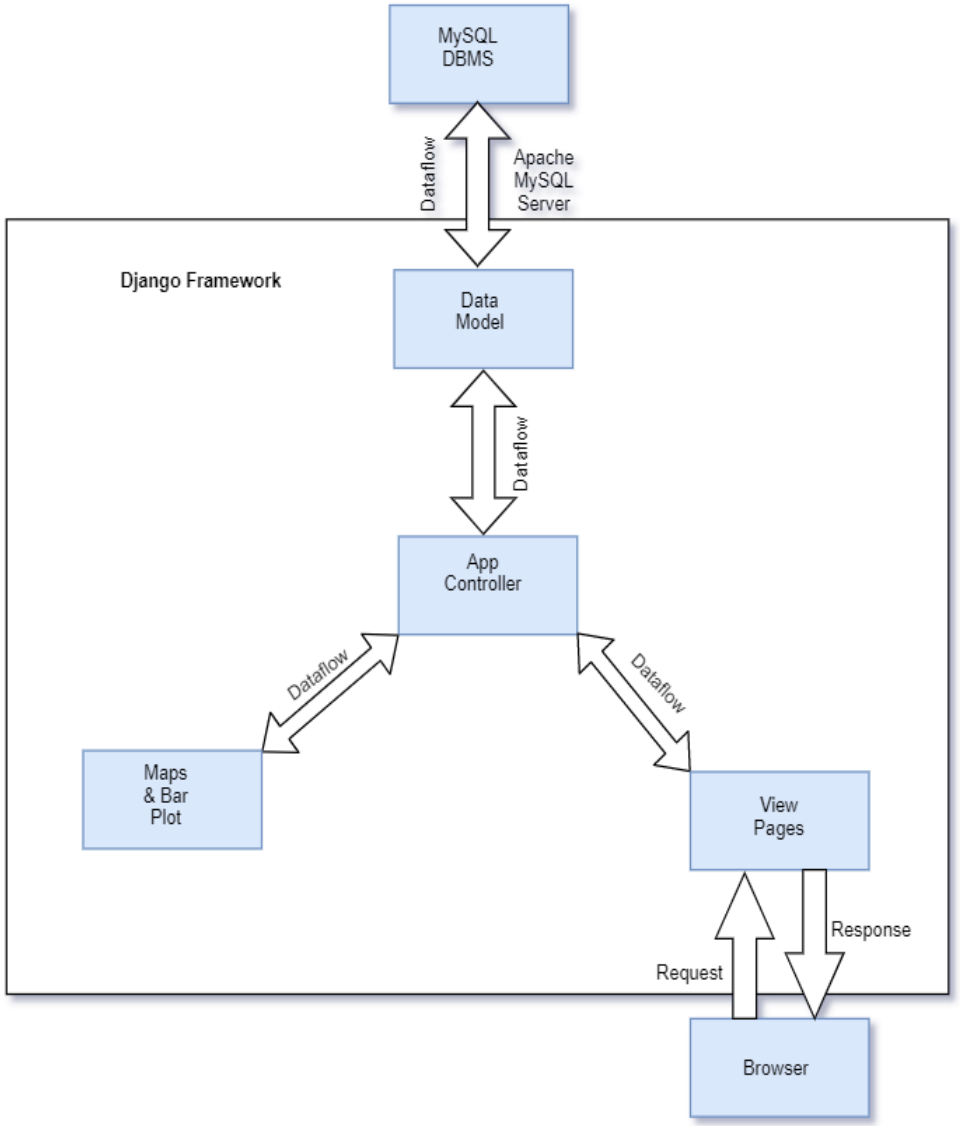

Fig-3.1

Our project uses Django framework for webapp integration and use of tools such as folium and Plotly for statistical representation of the current scenario in all seven provinces. The data has to be updated manually in MySQL database for the insurance of the authenticity. PyCharm IDE along with Django framework provides our local machine with development server capabilities. With the help of python controller programs, we can perform data operations on MySQL database through our webpage. Since, Plotly and folium both uses python dictionaries format for data input, MySQL datas are later converted into dictionary format having keys and values.

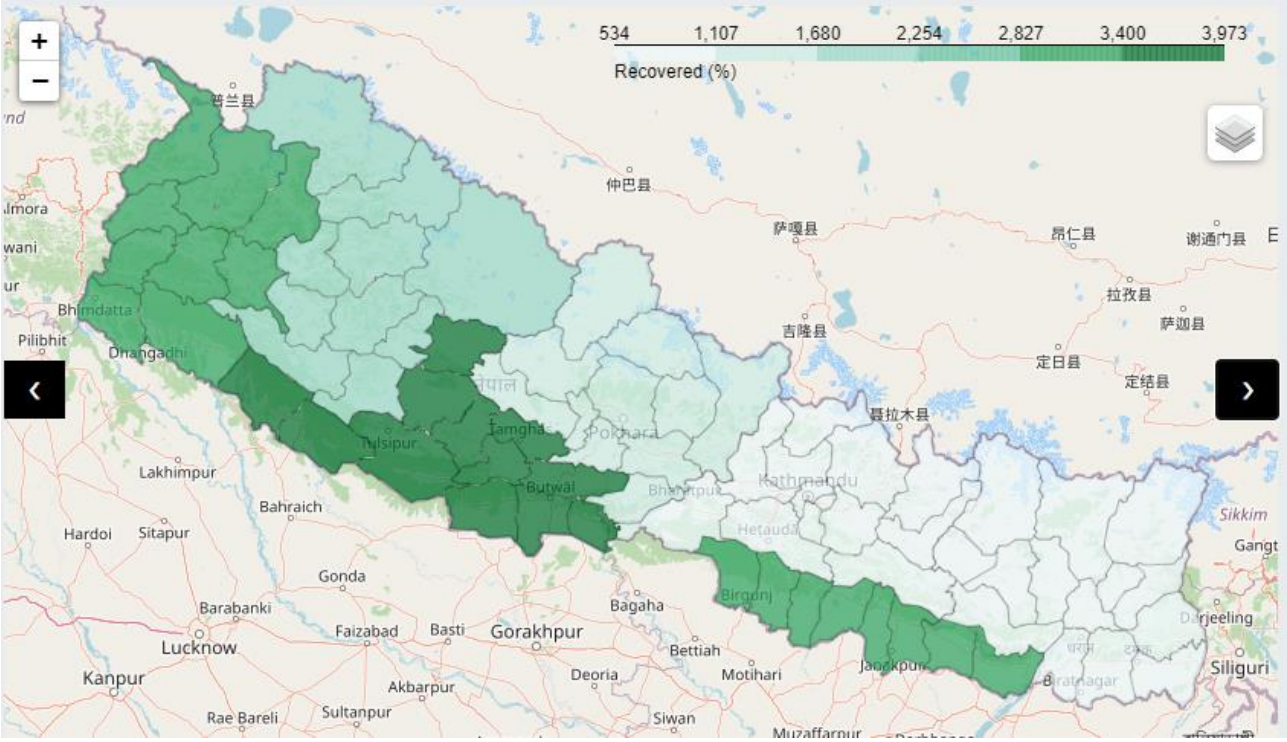

Fig-3.2

Fig 3.2 displays the population distribution of recovered patients throughout the province where darker areas represents higher recovery rates than that of lighter regions.

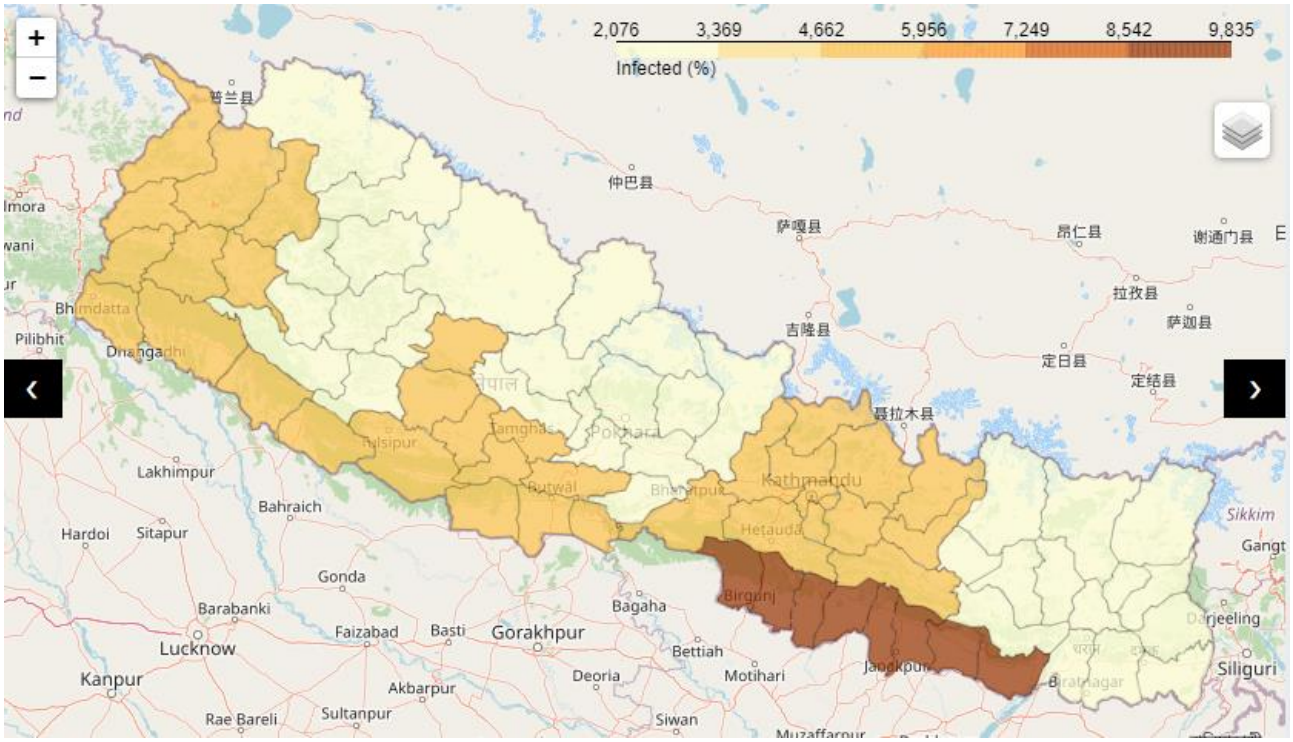

Fig-3.3

Fig 3.3 displays the population distribution of infected patients throughout the province where darker areas represents higher infected rates than that of lighter regions.

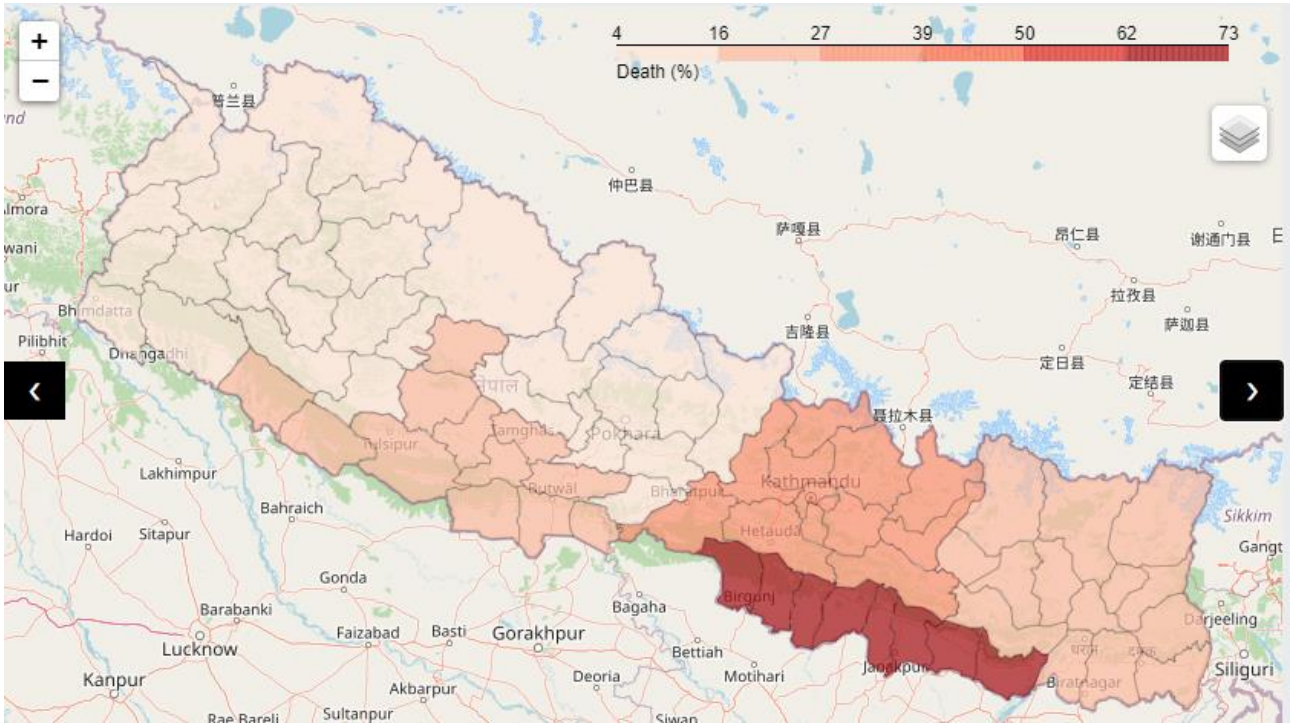

Fig-3.4

Fig 3.4 displays the population distribution of death rates throughout the province where darker areas represents higher death rates than that of lighter regions.

Any user visiting the webpage is presented with a homepage that has a choropleth map displayed all of which are created using folium and placed in a HTML/CSS slider, upon request from the browser. App controller creates map objects and is pushed into the view page where map object is then rendered into HTML and displayed as an interactive map with navigational features including a legend.

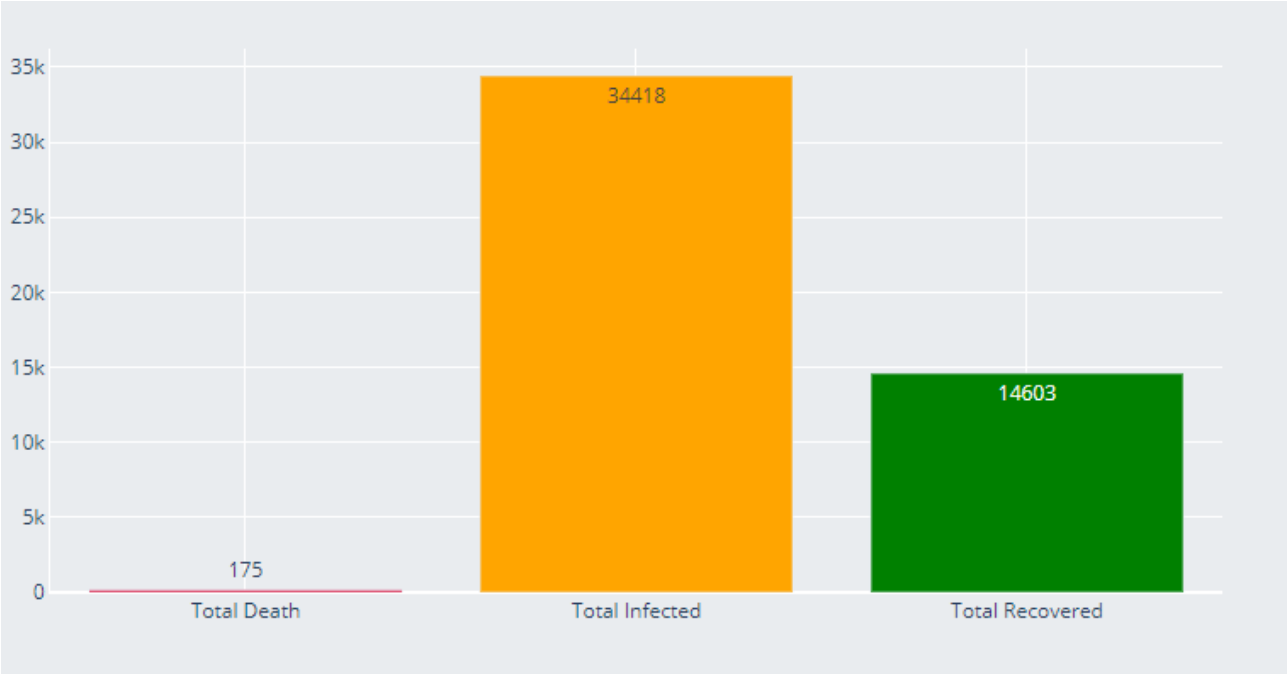

Fig-3.5

The bar diagram displayed above shows the total death, infected and recovery rates throughout all provinces where total death rate is represented using red color, infected rates are displayed using orange color and recovered rates are shown using green color.

Province	Infected	Death	Recovered
province_1	2947	20	858
province_2	9835	73	3164
province_3	5496	37	534
province_4	2076	8	1306
province_5	5950	24	3973
province_6	2402	4	1722
province_7	5712	9	3046

Add Cases

Fig-3.6

Fig 3.6 is a tabular representation of population of Infected, Death and Recovered patients. Here, the datas can also be edited, removed using edit, remove buttons respectively and also new cases can be added from the Add Cases button that redirects to the following form.

Fig-3.7

Using above form, new cases can be added for any of seven provinces using the dropdown menu.

3.1. System Requirement Specification

3.1.1. Software Specification

The web application ‘COVID Mapping’ was designed and developed using a combination of front-end and back-end tools. The backend of the project was developed in Django, a Python based free and open-source web framework developed by DFS. The database was developed using MySQL, while HTML and CSS were used as the main tools for frontend development. For more responsive, interactive and sophisticated design, Bootstrap 3.3.7 was used. For webapp integration we used tools such as Folium and Plotly for statistical bar representation of the states.

3.1.2. Hardware Specification

The web application was developed on a PC with Intel Core i7-5500U CPU processor and 8 GB of DDR3 RAM. Since the website is simplistic, sophisticated hardware is not required. Any modern PC capable of running a modern OS should suffice.

Chapter 4: Discussion on Achievements

None of us were familiar with the Python language nor the Django framework we chose to work with so the project came to life from point zero. On top of working with folium, bar plots, NumPy, and pandas, with pretty limited working time frame the learning curve was hard to trace. Nonetheless, with a huge python community on the internet we could basically surf to everything we needed to learn thus we did the same. During project setup, integrating Python environment was hindered by already existing Anaconda services. Later, the problem was solved by integrating the Conda environment instead of Python environment which provided services of both Python and Anaconda. Since, folium uses dictionaries format of data for plotting, we had to transform table data into pandas dictionaries and use NumPy to manipulate dictionaries to the likes of the available co-ordinates data provided by folium community for tracing provincial borders. We also wanted to include total number of PCR tests conducted in each province but the datas were changing in hourly basis so without a proper scraper, test results were hard to trace.

Features

- An all in one platform to show death, infected and recovered counts.
- User-friendly web app.
- Interactive choropleth map and bar plot.
- Manual data entry allows authentic data feeding.

Chapter 5: Conclusion and Recommendation

In this short span of time we have been able to collect all the required data and build the web map 'COVID- Mapping'. Building this project has been a great team work and every member have contributed to the fullest for the completion of 'COVID- Mapping'. The guidance from our supervisor helped us a lot to build this project. Further, we implemented the knowledge on Relational Database Management System which we'd been studying under the course titled 'COMP 232' thus we did the same using MySQL. Furthermore, including libraries like pandas, NumPy also which we're currently studying under the course titled 'MCSC 202'. Thus, with no doubts we can confidently claim we gained knowledge to our objectives hence the project work as presented.

5.1. Limitations:

Although Choropleth maps give a good visual impression of changes over a space/boundary, there are certain disadvantages to using them.

- Currently we have not acquired a domain hence our functionality is limited to a localhost.
- Manual data entry wouldn't be as efficient as scraping data.
- The UI could be more user-friendly with some added extensions and features.

5.2. Future Enhancement:

As an improvement to our project a robust scraper that would fetch datas from authentic sources could be added. Overcoming the limitations of a localhost, premium server features by co-operation with hosting companies could commercialize the web app. Also, number of tests performed in each state as per MoHP can be added in upcoming versions.

References

Alimetov, A. (n.d.). *worldometer*. Retrieved from https://www.worldometers.info/coronavirus/

Boston Children's Hospital. (2020). Retrieved from HealthMap: https://healthmap.org/en/

JHU. (2020). *Johns Hopkins University & Medicine*. Retrieved from https://coronavirus.jhu.edu/map.html

NITAG. (n.d.). *European Centre for Disease Prevention and Control (ECDC)*. Retrieved from https://www.ecdc.europa.eu/en/geographical-distribution-2019-ncov-cases

Appendix

Table1: Gantt Chart

Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Research and study	■	■	■											
Graphic Designing				■	■	■								
Core programming						■	■	■	■	■				
Program testing											■	■		
Documentation													■	■