

KATHMANDU UNIVERSITY

Department of Computer Science and Engineering
Dhulikhel, Kavre



A Project Report
on
"Covid-19 Situation Analysis Application"

[Code No. : COMP 206]

(For partial fulfillment of 2nd Year/ 3rd Semester in Computer Science)

Submitted by
Aayush Pokharel (Roll No. 43)
Dikshya Poudel (Roll No. 45)
Sajag Pradhanang (Roll No. 47)
Samjhana Bhusal (Roll No. 60)

Submitted to
Project Coordinator
Nabin Ghimire
Dept of Computer Science and Engineering

Submission Date: 13th August, 2021

Bonafide Certificate

This project work on “Canteen Management System” is the
bonafide work of

”

Aayush Pokharel(Roll No. 43)
Dikshya Poudel (Roll No. 45)
Sajag Pradhanang (Roll No. 47)
Samjhana Bhusal (Roll No. 60)

”

who carried out the project work under my supervision.

Project Supervisor

Name: Santosh Khanal

Assistant Professor

Department of Computer Science and Engineering

Acknowledgement

We wish to express our sincere thanks to the Department of Computer Science and Engineering for including the COMP 206 project into our curriculum. We would like to express our heartfelt gratitude to our project coordinator Mr. Nabin Ghimire and our supervisor Mr. Santosh Khanal for their regular guidance and encouragement throughout the project. Taking this opportunity, we would like to thank all those individuals who directly or indirectly helped us in making this project successful one be it by encouraging us throughout the project or else through their valuable suggestions which we have tried our best to assimilate within our work.

Abstract

The project ‘Covid-19 Situation Analysis Application’ proposal is drafted to meet the prerequisites to partially fulfill the COMP 206 course offered by the Department of Computer Science and Engineering at Kathmandu University and to cover for the pitfall of previously proposed Project ‘Portfolio Management and Finance Management Application’ . This project is designed to overcome the lack of centralized facility to deeply analyse COVID-19 situational data on international and national level. We, the involved project members, have decided to create a compact application that can run on Windows platform which scrapes data from multiple sites like worldometer.info, mohp.gov.np, covidnepal.org, covid19.wh.int allowing us to work on the dataset and plot various diagrams and charts necessary for in depth demographic analysis. It will allow customizing the charts based on demographic factors and also present trends before, during and after a major wave. We are accomplishing this project by using dynamically typed object oriented programming language(Python) and Qt framework to handle the GUI portion of the application, Selenium and Scrapy to scrape web data, pandas for handling dataset and matplotlib to finally plot the data. The main goal of this project is to develop an efficient set of python code that can stand up to real world scenarios while increasing the programming skill of involved project members in hope of tackling harder projects in the forthcoming days.

Keywords: Python,Qt framework, Selenium, Scrapy

Contents

1	CHAPTER 1: INTRODUCTION	1
1.1	Background	1
1.2	Objectives	1
1.3	Motivation and Significance	1
2	CHAPTER 2: RELATED WORKS	3
2.1	worldometer.info	3
2.2	covid19.mohp.gov.np	4
2.3	covidnepal.org	4
3	CHAPTER 3: DESIGN AND IMPLEMENTATION	5
3.1	Architectural Design	5
3.1.1	Flow Chart	5
3.1.2	Use Case Diagram	7
3.2	System Requirement Specification	8
3.2.1	Software Requirements:	8
3.2.2	Hardware Requirements:	8
4	CHAPTER 4: DISCUSSION AND ACHIEVEMENT	9
4.1	Methodology	9
4.1.1	Research and Study	9
4.1.2	UI design and Prototyping	9
4.1.3	Core Programming	9
4.1.4	Program Testing	10
4.2	Features	11
4.2.1	Web Scraping	11
4.2.2	GUI	11
5	CHAPTER 5: CONCLUSION AND RECOMMENDATION	12
5.1	Limitations	12
5.2	Further Enhancements	12
6	Refrences	13
7	Appendix	14

List of Figures

1	Worldometer Website	3
2	Ministry of Health and Population Website	4
3	CovidNepal.org Website	4
4	SplashScreen FlowChart	5
5	MainScreen FlowChart	6
6	UseCase Diagram	7
7	Splash Screen	14
8	Main Window - Dashboard	15
9	Gant Chart	16

1 CHAPTER 1: INTRODUCTION

1.1 Background

COVID-19 pandemic is a global problem which has led to a dramatic loss of human life worldwide and presents an unprecedented challenge to public health, food systems and the world of work. It has stifled the global economy, caused untold amounts of grief among family members and friends, and forced people into social distancing and lockdown as means of virus transmission prevention.

Compared to some other countries, Nepal was still in a better situation during the first wave of the pandemic but during the second wave, it was hit quite worse. The lack of accessible and accurate information have caused people to be unaware of Covid-19 risks and the need to protect themselves, get tested and vaccinated. The crisis, illness, death and isolation have increased the vulnerability of the population as a whole.

There are few available sites that publish information on the covid situation but any data required for an in depth analysis based on different demographic factors are needed to be sourced from different websites. The lack of standardization of data between different websites causes cross referencing to be very difficult. To challenge this difficulty, we are aiming to make an open source program that is available to everyone with no cost overhead for operation. This system will be used by end users for the purpose of analyzing the COVID-19 situation on an international as well as on national level based on various demographics. Furthermore, this application shall also report on the availability of medical infrastructures available in Nepal to COVID-19 patients based on geographical region.

1.2 Objectives

The main objective of this project is to understand the intermediate concepts of dynamically typed language and web scraping for enhancing our knowledge to solve real life problems using an Object-oriented paradigm. Besides, the other objectives are listed below:

- To make various configurable graphs of multiple demographic indicators to facilitate technical analysis.
- To make an app which can keep people updated about the COVID-19 situation at local, national and international level.
- To create a database that allows user to know about the current medical facilities near them.

1.3 Motivation and Significance

Our project is particularly inspired by various sites like Worldometer and covid19.mohp.gov.np. This program is being designed with an aim of collecting data that are available on dif-

ferent sites with our prior experience of the struggle of getting information in an intuitive manner. This will present a relevant user interface which provides easy, enjoyable and effective interaction between the user and the application along with the explanation of various graphs and their trends. Most notably, it will allow for analysis of transmission and mortality trends with cross reference to health infrastructures via line graph, making it more accessible for in depth understanding of the pandemic situation to a person untrained in technical analysis. Furthermore, this program presents features such as Vaccine development situations, local governmental updates, etc to facilitate understanding about the pandemic among users as a means to combat its spread.

2 CHAPTER 2: RELATED WORKS

There are similar sites available which are being used globally as well as nationally which are being operated by different NGO and governmental organizations and to keep track of the COVID-19 pandemic.

Some of such sites that are in use are:

2.1 worldometer.info



Figure 1: Worldometer Website

Worldometer is a website which manually analyzes, validates, and aggregates data from thousands of sources in real time and provides global COVID-19 live statistics for a wide audience of caring people around the world. It's sources include official websites of Ministries of Health or other Government Institutions and Government authorities' social media accounts. It collects and processes data around the clock, 24 hours a day, 7 days a week and performs multiple updates per minute.

2.2 covid19.mohp.gov.np

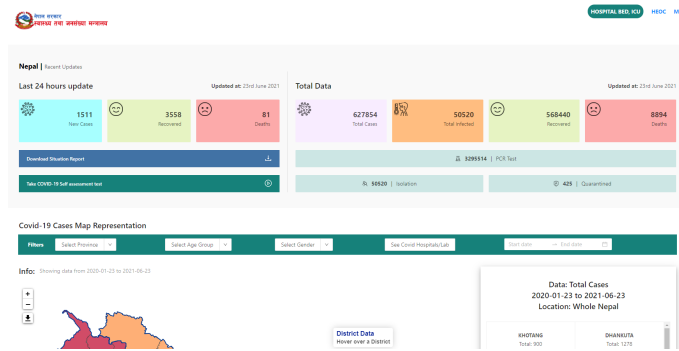


Figure 2: Ministry of Health and Population Website

It is an official website of Ministry of Health and Population (MoHP) designed to provide authentic information about covid situation in Nepal. It consists of all the covid related data like active cases, death cases, recovered cases classified on the basis of district, province, age, sex and compares them. It also provides information about the hospital bed availability for different regions to help people find proper care in time.

2.3 covidnepal.org

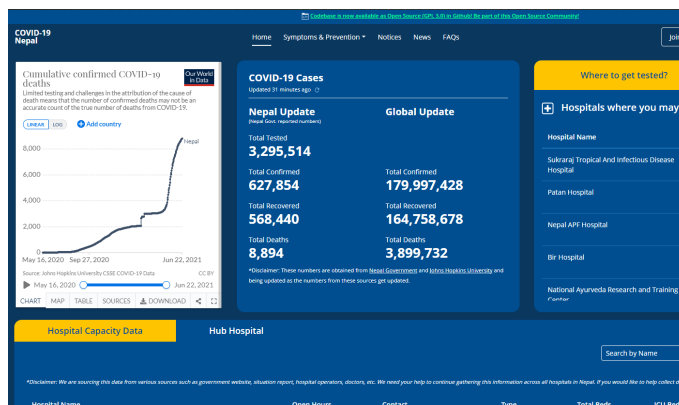


Figure 3: CovidNepal.org Website

It is an open source platform that provides reliable information about Covid-19 in Nepal along with global cases update. It also provides a general information about where and when to get tested and covid hospitals for a particular region in Nepal to get admitted along with the information about its symptoms and prevention methods.

3 CHAPTER 3: DESIGN AND IMPLEMENTATION

3.1 Architectural Design

3.1.1 Flow Chart

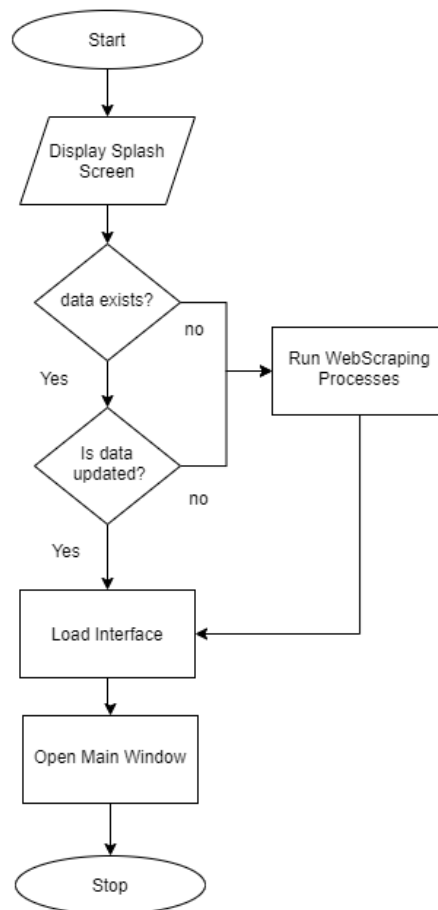


Figure 4: SplashScreen FlowChart

This flowchart shows the inner working of the Splash Screen Logic of the program. It checks if various data files are present and are upto date. If it doesn't find it, it runs the necessary Webscraping processes to scrape the data.

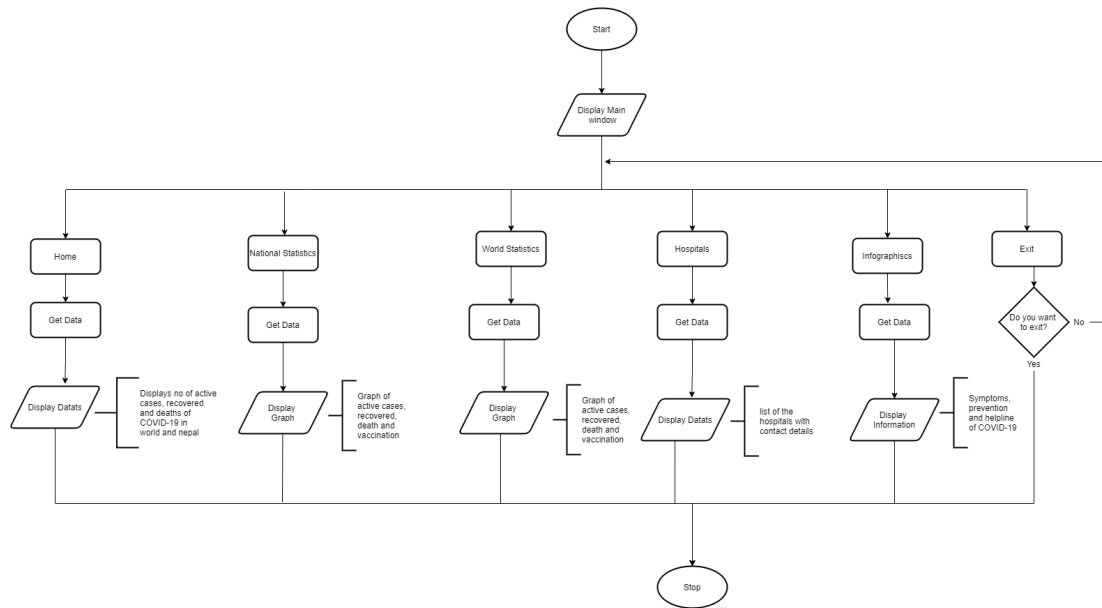


Figure 5: MainScreen FlowChart

This flowchart shows the inner working of the Main Window and the logic of the program when user interacts with the GUI. It displays and updates graphs, charts, tables based on filters configured by the user.

3.1.2 Use Case Diagram

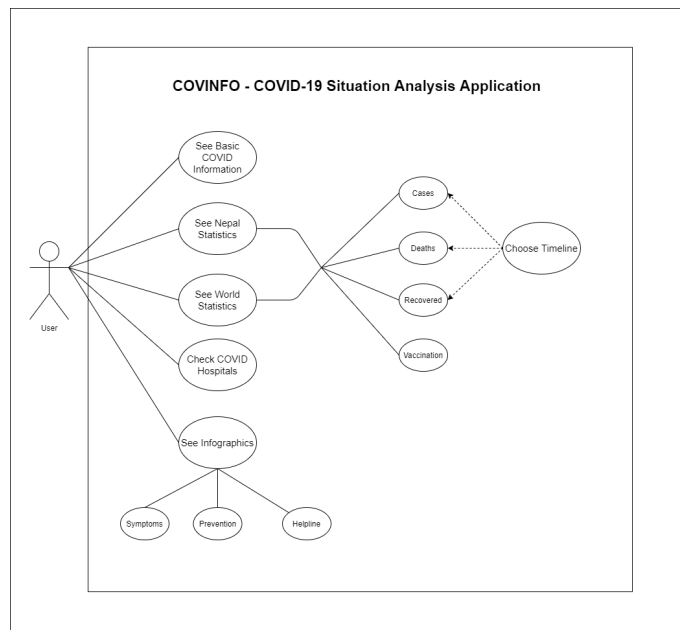


Figure 6: UseCase Diagram

This use case diagram illustrates the various UI elements and its functions on Main Window of the application.

3.2 System Requirement Specification

3.2.1 Software Requirements:

- **Front End Tools**

- The program is created using Python.
- Development Tools : QT frameworks

- **Back End Tools**

- Selenium and Scrapy modules used for scraping web information.
- Pandas module used to make datasets.
- Matplotlib and QTGrapph used for plotting.
- Web Browser supporting Selenium WebDriver (*Here, Google Chrome v92.0.4515.131*)

3.2.2 Hardware Requirements:

- **Compatibility:** Compatible with all PCs running on Windows platform.

Python : A dynamically typed multi paradigm general purpose programming language.

QT frameworks : An open source cross platform multi language widget toolkit that helps in developing GUI Application.

4 CHAPTER 4: DISCUSSION AND ACHIEVEMENT

4.1 Methodology

The whole process of developing the software has been divided into following aspects:

- Research and study
- UI Design and prototyping
- Core programming
- Program Testing
- Documentation

4.1.1 Research and Study

This project being the first of the kind we have ever done, we decide to choose a subject matter that we clearly know and have experience with ie. the plotting graph using matplotlib. We have interacted with the various software/sites mentioned above in chapter 2 on a semi-regular basis and have experienced its strength and pitfall. We are trying to compile the information on nation and international stage and present it on a singular platform.

4.1.2 UI design and Prototyping

The initial designs are sketched on A4 papers by our members and shared on group chat for peer review and then use QT Creator's design tab to stub test the look of our UI before adding any functionality to the various widgets.

4.1.3 Core Programming

The most amount of project time is going to be spent in this stage. We are using the Python language to write our program. QT framework provides the necessary tools to create the GUI. Selenium is being used to surf the web and get html of webpages and scrapy is being used on the html file to parse and extract data which is converted into datasets via Pandas for easy import and manipulation as per end user requirement; for instance, getting data of certain dates only. Finally the QTGraph and matplotlib library is used to plot the dataset.

4.1.3.1 Issues Encountered

- ***Change of project halfway***

The project was after the initial project was not upto par with our Supervisor. Since more than half of the work was already completed in the initial project, starting anew on a different topic was difficult to adjust.

- ***Deciding which UI language to use (QML or Qt XML)***

Due to the confusing nature of multiple libraries providing python binding for QT, we had problems choosing the approach to build the User interface. Initially we tried with QML to have dynamic and easily configurable UI for easier iteration but due to different OS platform present in team members' PC, getting there was version mismatch of the available library between each team members. Next we tried with PySide6 but due to lack of support for Matplotlib embedding, we ultimately had to choose PySide2.

- ***Python library Setup***

Due to lack of proper experience in working with Command Line Interface, Team members needed to be brought up to knowledge regarding python virtual environments and pip along with teaching relevant commands in Powershell and Bash.

- ***Brainstorming features to be added***

Initially the project was on different trajectory and due to amount of time invested, team members were locked in a certain direction. When the project topic was changed, the team had great difficulty in coming up with new ideas.

4.1.4 Program Testing

Unit testing was done as soon as we completed the code of a single widget to check its functioning properly. This was repeated multiple times during the development period to create a robust system and once the core programming is completed, alpha testing was done to find different types of bugs or ill optimised issues.

4.1.4.1 Bugs found and Debugged

- Graph wasn't being updated when the timeline was being changed.
- Program was loading old data files in Splash screen instead of running scraping command.
- Scrapy wasn't making files in proper location after scraping was complete.
- Blank information was being scraped from the website of MOHP until the issue was identified.

4.2 Features

To make a robust application, alot of features were added to the program. Both Backend Webscraping and Frontend GUI have access to a wide array of features and some of them are as follows.

4.2.1 Web Scraping

- WebPages were controlled through the use of Selenium WebDriver.
- Scrapy was used to scrape data from webpages.

4.2.2 GUI

- Basic COVID statistics of Nepal and World.
- Custom graphical representation of chosen timeline
- Information regarding available COVID hospitals in the country
- Infographics to know the symptoms, prevention for the virus with helpline

5 CHAPTER 5: CONCLUSION AND RECOMMENDATION

With the constant supervision of the supervisor and hard work of team members, the project is finally complete. After this project we believe our team is ready to tackle more sophisticated projects in the future.

5.1 Limitations

The program has the following limitations:

- ***Making Covid Cases Predictions***

The project supervisor suggested to make a feature to make a prediction for the future as per current data. The initial idea was for the team to get an average of cases each months and calculate deviation. But the team couldn't have a proper idea to solve it and due to the rush and other obligations the feature was scraped to focus on other features.

- ***Selenium WebDriver***

The project uses Selenium WebDriver so the Browser must match the WebDriver available. This can easily be solved by the end user by copying the webdriver file for their Browser into the relevant directory from the web.

- ***Making the project's Executable file***

The documentation for creating a .exe file from a python script was confusing and resources available online pointing to different and confusing solutions. Since it's easy enough to create a virtual environments and run a python script in it. This step was ignored.

5.2 Further Enhancements

The following action could be undertaken to improve the program.

- ***Making a proper prediction model using Machine Learning***

None of the team members had any experience with Machine Learning and so were unable to make a proper data prediction functionality. In the next Semester, if given time the team members could focus on learning ML and working to make a prediction model.

- ***Making an Executable file***

Due to the confusing nature of the the process and the ease of just running a python script via the command line. Importance wasn't given to make a executable file (.exe) due to the time constraints and different OS present in team members' PC.

6 References

Worldometer.(n.d.). COVID-19 CORONAVIRUS PANDEMIC.
<https://www.worldometers.info/coronavirus/>

Government of Nepal Ministry of Health and Population. (n.d.). <https://covid19.mohp.gov.np/>

COVID-19 Nepal. (n.d.) <https://covidnepal.org/>

7 Appendix

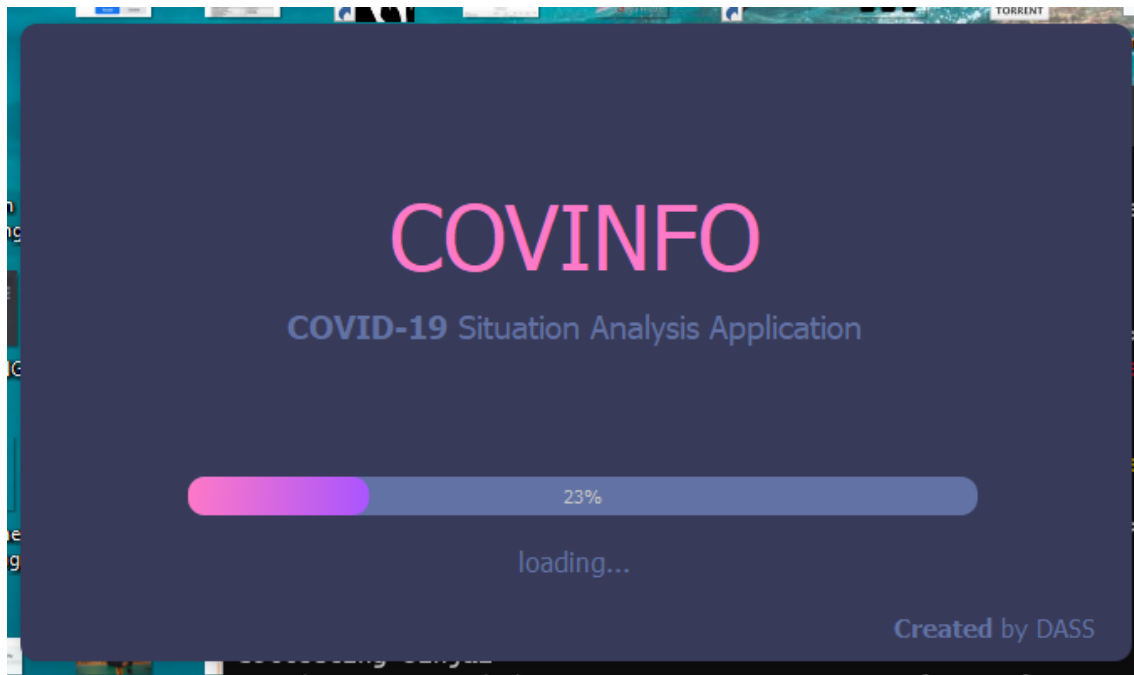


Figure 7: Splash Screen

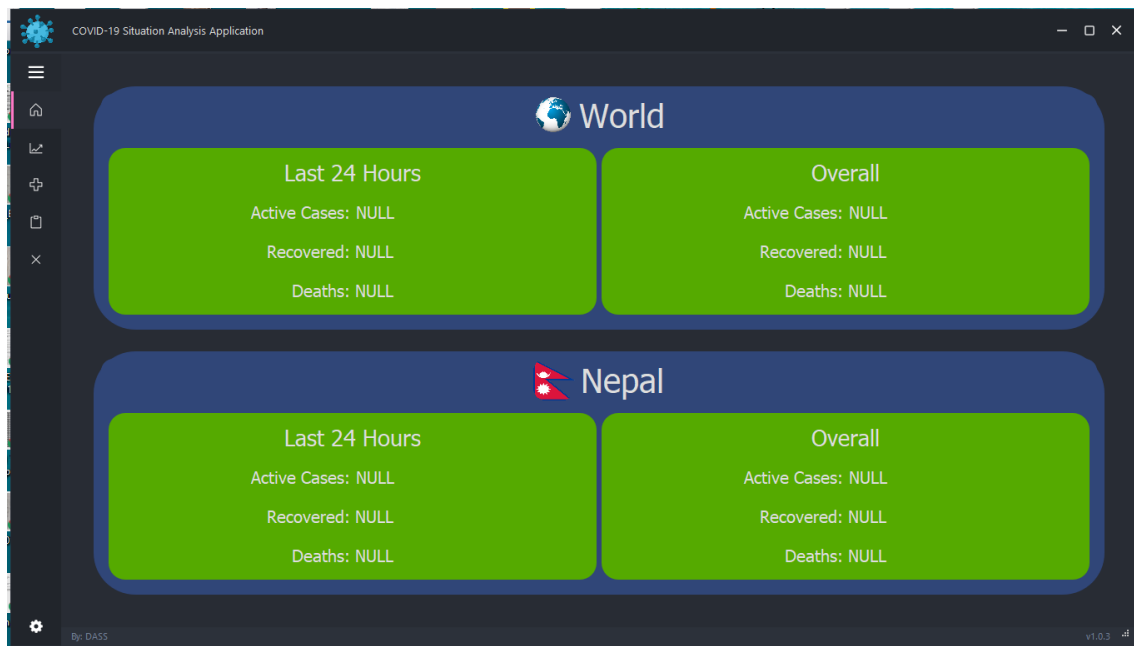


Figure 8: Main Window - Dashboard


Task Summary	Weeks					
	1	2	3	4	5	6 
Planning						
Design						
Coding						
Testing						
Documentation						

Figure 9: Gant Chart

Index



Task Completed