Big Data Architecture and Governance

Group Project | COVID 19 Infection Data

Navaneeta Naik | Nikunj Doshi | Yu Ren

Northeastern University
College of Engineering

PROJECT DETAILS

TEAM MEMBERS

Northeastern University
College of Engineering

NIKUNJ DOSHI :
PROJECT MANAGER

YU REN:
DATA ENGINEER

NAVANEETA NAIK:
QA & DATA ANALYST

KH :
END/ BUSINESS USER

VIVIDHA SINGH :
SPONSOR

PROJECT PLAN

- Our aim was to analyze and figure out the impact of Covid Cases in the different parts of the globe and how death cases and confirmed cases factored in driving of Covid cases.

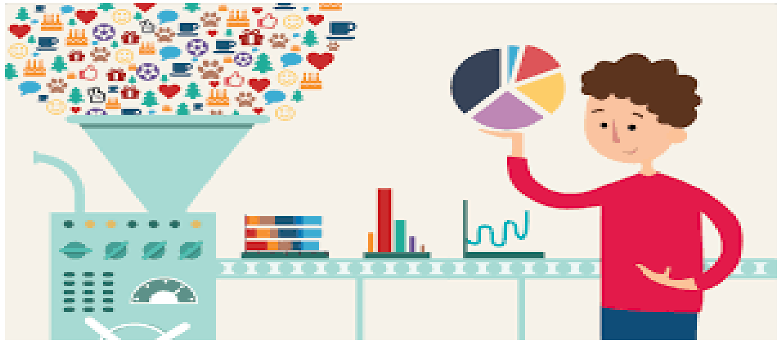
❑ How many Deaths occurred after Confirmed Cases?

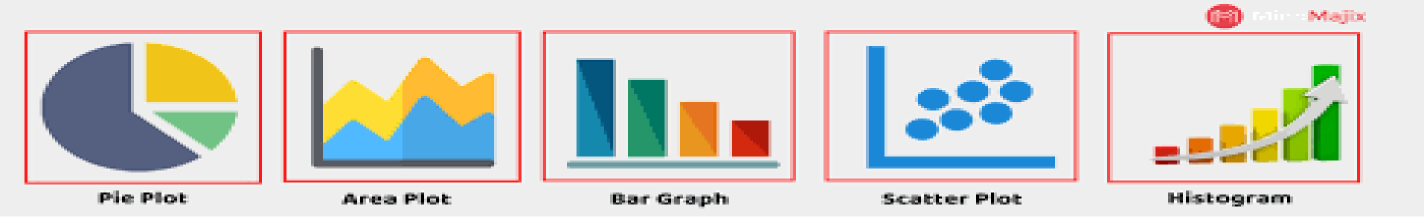
❑ Which countries showed highest Recovery?

❑ Was there any country who did not had any deaths after confirmed case?

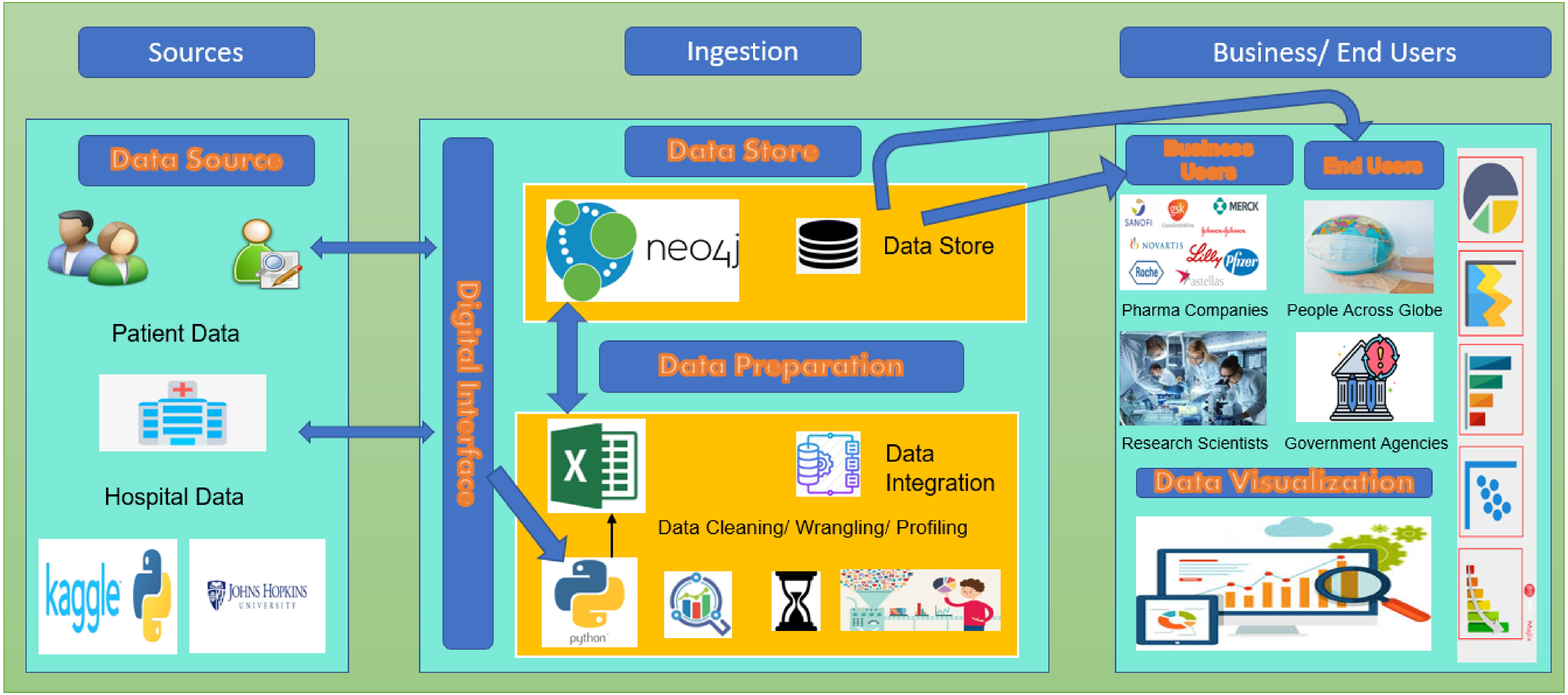
- With the help of this data set we would love to see more conclusions drawn so that with the help of our analysis, Business users like research scientists and Pharma Companies for creating vaccines,
- Also End Users like people all over the globe and Government Agencies who could draw some insights which may help them to improve their plans and guidelines.

Tools & Techniques Used





Vision Diagram



Sources

Ingestion

Business/ End Users

Data Source

Patient Data

Hospital Data

Digital Interface

Data Store

Data Store

Data Preparation

Data Integration

Data Cleaning/ Wrangling/ Profiling

Business Users

End Users

Pharma Companies

People Across Globe

Research Scientists

Government Agencies

Data Visualization

Business / End Users

Business Users

Pharma Companies

Research Scientists

End Users

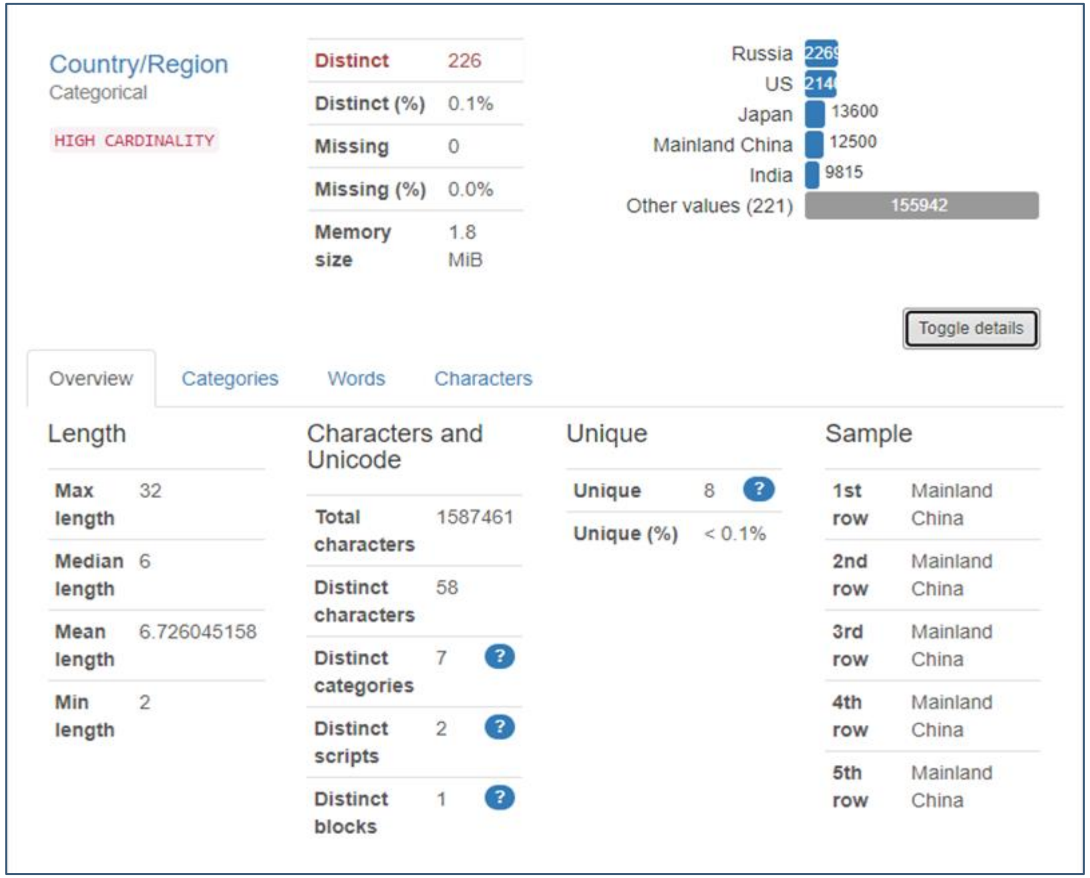
People Across Globe

Government Agencies

DATA PRE PROCESSING

Data Profiling - Python

- Data Profiling provides summarized information about our dataset
- Provides information about each column- Missing values, Duplicates, Zeros and Unique columns
- The profiling provided information about the missing values in our dataset.



Country/Region
Categorical

HIGH CARDINALITY

Distinct	226
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.8 MiB

Russia	2269
US	214
Japan	13600
Mainland China	12500
India	9815
Other values (221)	155942

Length

Max length	32
Median length	6
Mean length	6.726045158
Min length	2

Characters and Unicode

Total characters	1587461
Distinct characters	58
Distinct categories	7 ?
Distinct scripts	2 ?
Distinct blocks	1 ?

Unique

Unique	8 ?
Unique (%)	< 0.1%

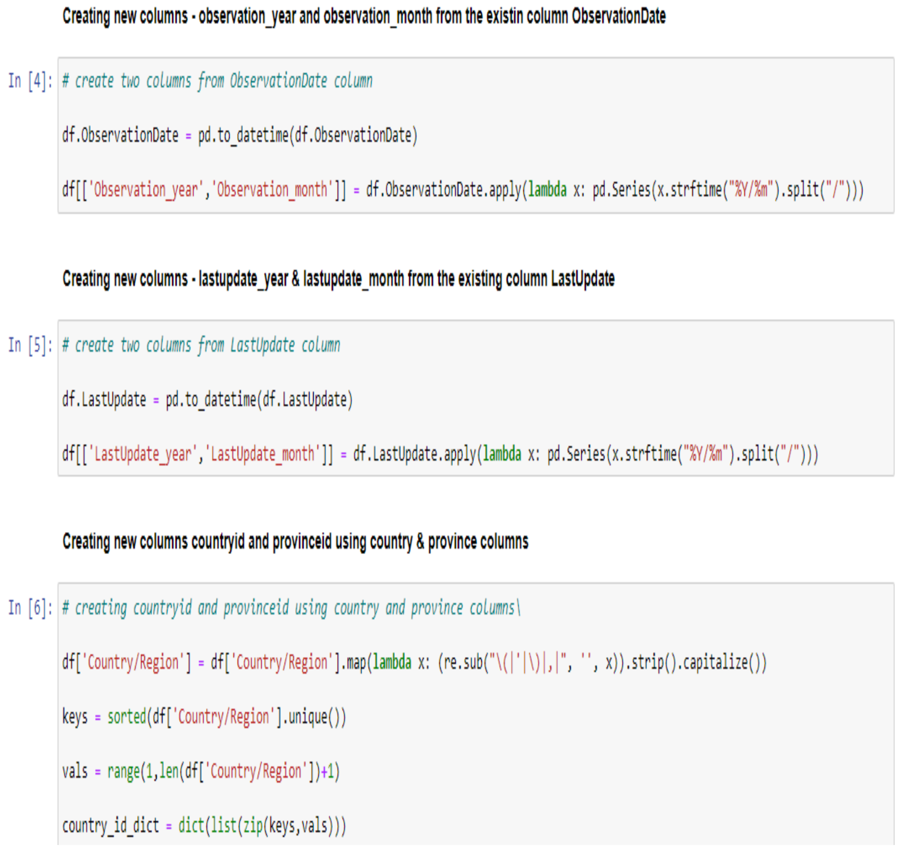
Sample

1st row	Mainland China
2nd row	Mainland China
3rd row	Mainland China
4th row	Mainland China
5th row	Mainland China

Data Wrangling - Python

Northeastern University
College of Engineering

- Data wrangling is the process of transforming data to make it more appropriate and valuable to be used in analytics
- For analytics purpose, we created new columns Observation_month & Observation_year from ObservationDate and created LastUpdate_month & LastUpdate_year from LastUpdate
- Two additional columns were created ProvinceID, Country_Id to identify distinct countries and provinces within those countries



Creating new columns - observation_year and observation_month from the existin column ObservationDate

```
In [4]: # create two columns from ObservationDate column

        df.ObservationDate = pd.to_datetime(df.ObservationDate)

        df[['Observation_year','Observation_month']] = df.ObservationDate.apply(lambda x: pd.Series(x.strftime("%Y/%m").split("/")))
```

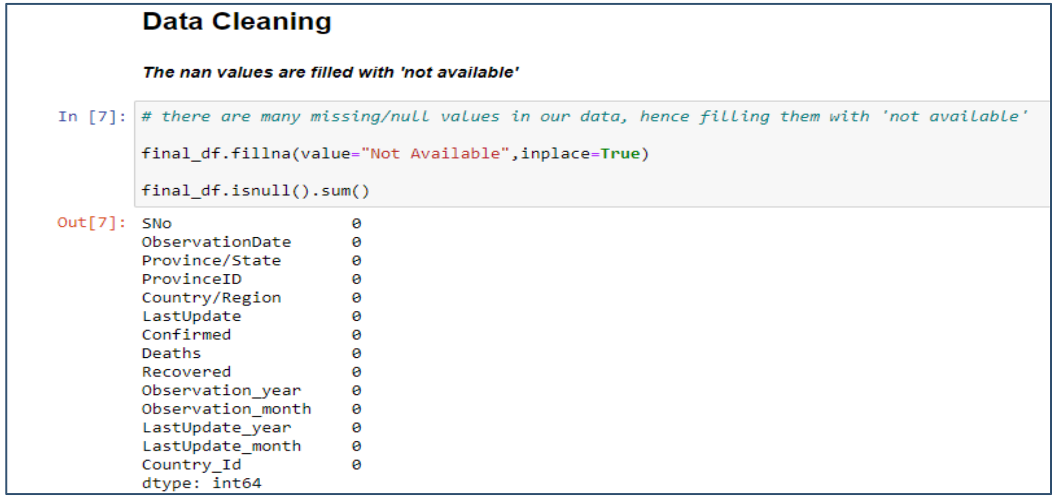
Creating new columns - lastupdate_year & lastupdate_month from the existing column LastUpdate

```
In [5]: # create two columns from LastUpdate column

        df.LastUpdate = pd.to_datetime(df.LastUpdate)

        df[['LastUpdate_year','LastUpdate_month']] = df.LastUpdate.apply(lambda x: pd.Series(x.strftime("%Y/%m").split("/")))
```

Creating new columns countryid and provinceid using country & province columns

```
In [6]: # creating countryid and provinceid using country and province columns\

        df['Country/Region'] = df['Country/Region'].map(lambda x: (re.sub("\(|'|\)|,|", '', x)).strip().capitalize())

        keys = sorted(df['Country/Region'].unique())

        vals = range(1,len(df['Country/Region'])+1)

        country_id_dict = dict(list(zip(keys,vals)))
```

Data Cleaning

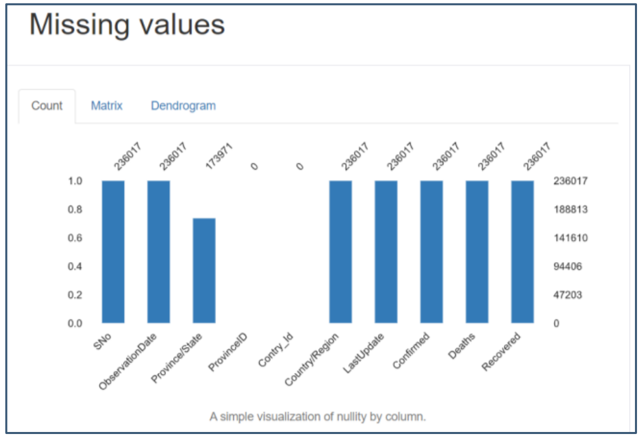


Data Cleaning

The nan values are filled with 'not available'

```
In [7]: # there are many missing/null values in our data, hence filling them with 'not available'

        final_df.fillna(value="Not Available",inplace=True)

        final_df.isnull().sum()

Out[7]: SNo                    0
        ObservationDate        0
        Province/State         0
        ProvinceID             0
        Country/Region         0
        LastUpdate             0
        Confirmed              0
        Deaths                 0
        Recovered              0
        Observation_year       0
        Observation_month      0
        LastUpdate_year        0
        LastUpdate_month       0
        Country_Id             0
        dtype: int64
```

Missing values

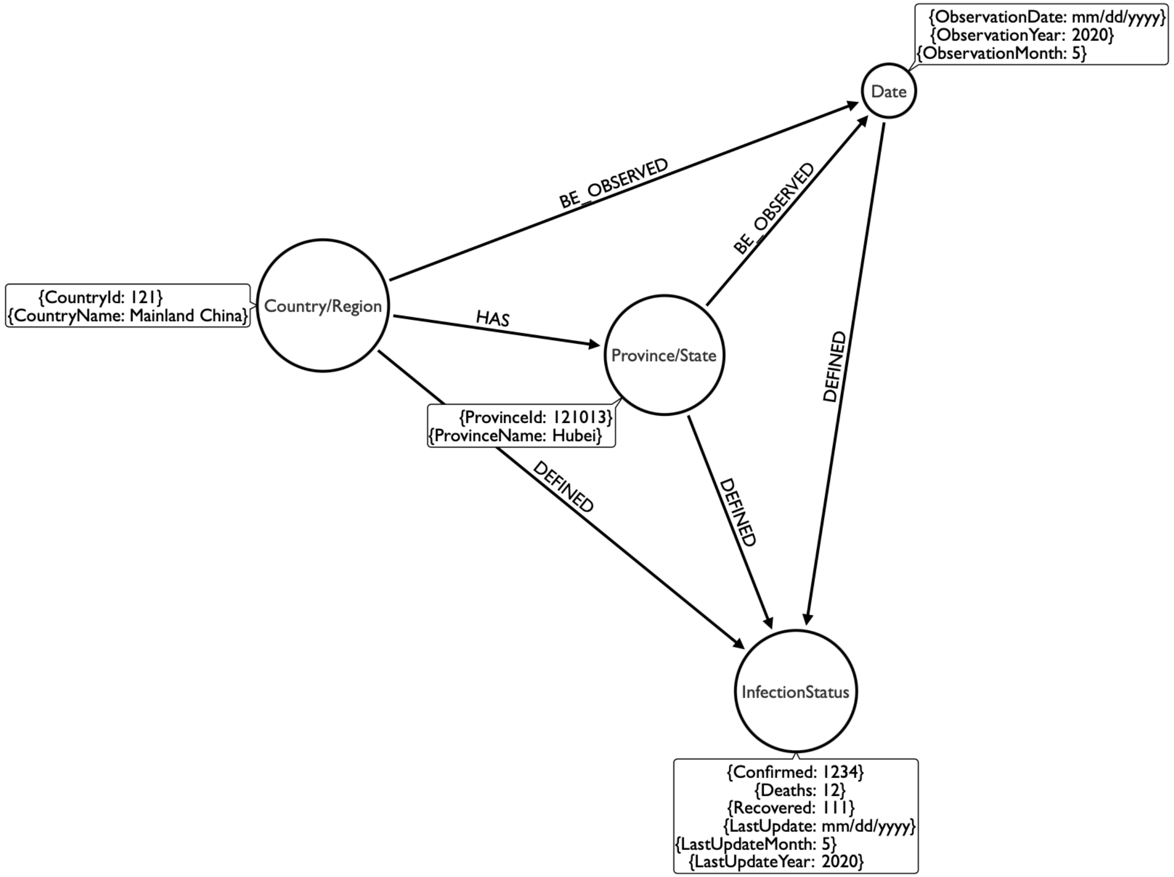


A simple visualization of nullity by column.

- There were missing values in the column State/Province
- Replaced null values in State/Province with ‘not available’

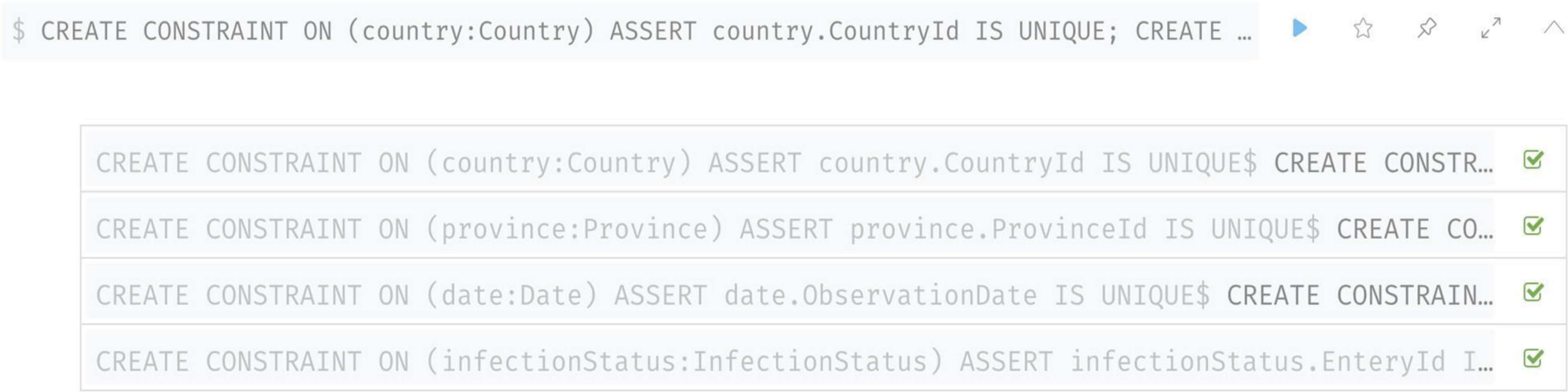
DATA LOADING – NEO 4J

NEO4J – Data Modal



NEO4J Screenshot - Create Constraints

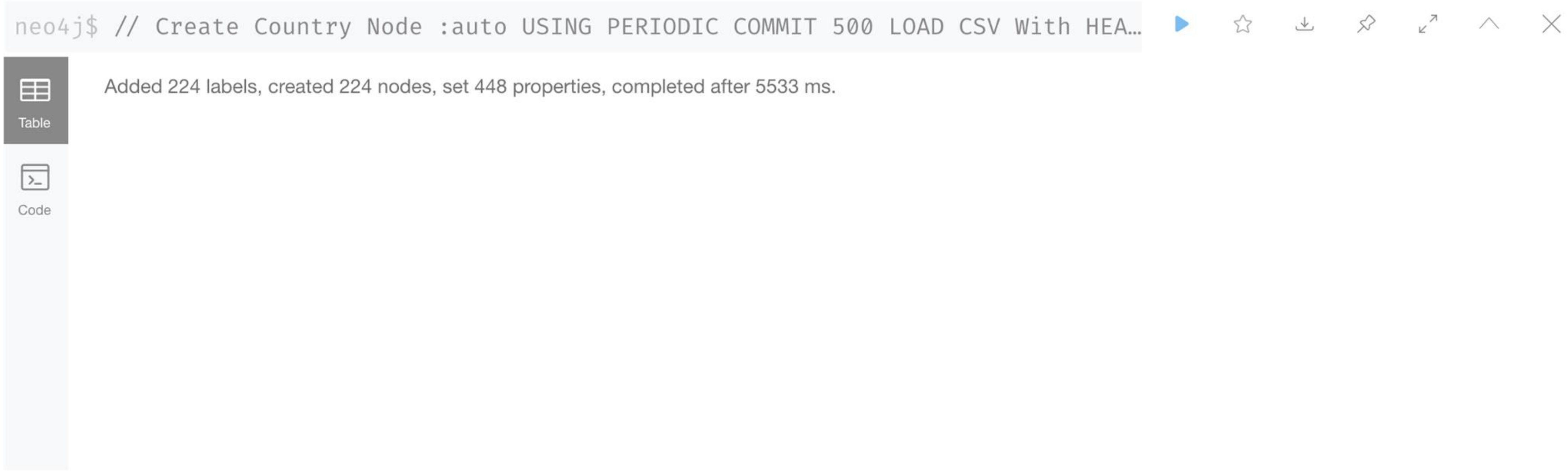
Create Contraints: country(CountryId), province (ProvinceId), date(ObservationDate), infectionStatus(EntryId)



```
$ CREATE CONSTRAINT ON (country:Country) ASSERT country.CountryId IS UNIQUE; CREATE …
```

```
CREATE CONSTRAINT ON (country:Country) ASSERT country.CountryId IS UNIQUE$ CREATE CONSTR…  ☑
CREATE CONSTRAINT ON (province:Province) ASSERT province.ProvinceId IS UNIQUE$ CREATE CO…  ☑
CREATE CONSTRAINT ON (date:Date) ASSERT date.ObservationDate IS UNIQUE$ CREATE CONSTRAIN…  ☑
CREATE CONSTRAINT ON (infectionStatus:InfectionStatus) ASSERT infectionStatus.EnteryId I…  ☑
```

NEO4J Screenshot - Create Nodes

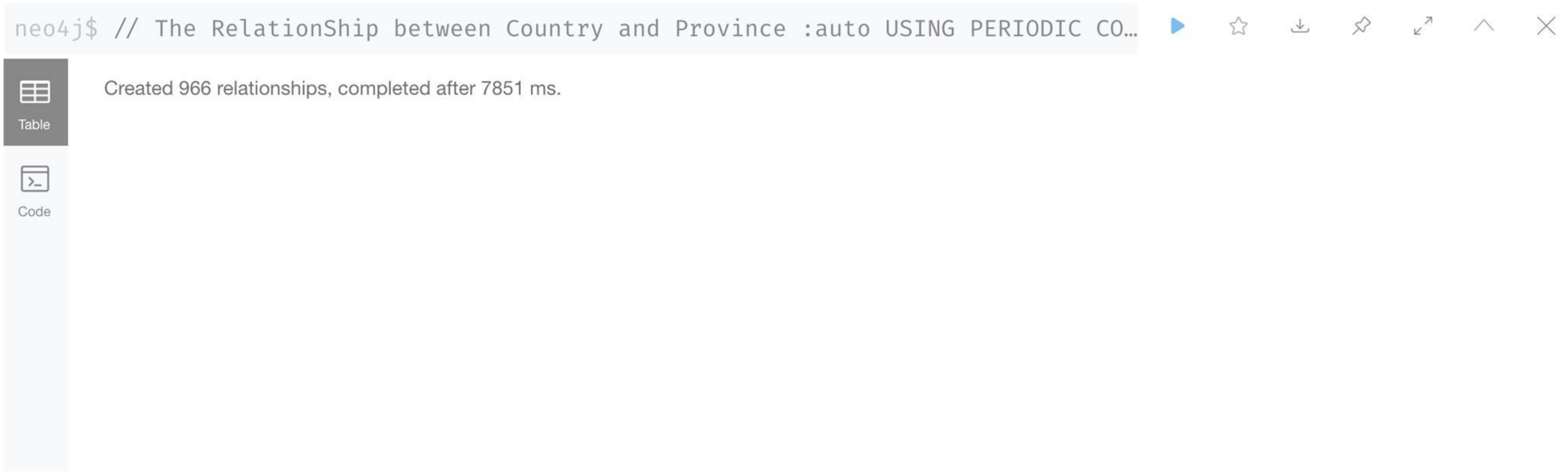


```
neo4j$ // Create Country Node :auto USING PERIODIC COMMIT 500 LOAD CSV With HEA…
```

```
Added 224 labels, created 224 nodes, set 448 properties, completed after 5533 ms.
```

Added 224 labels, created 224 nodes, set 448 properties, completed after 5533 ms.

NEO4J Screenshot - Create Relationships

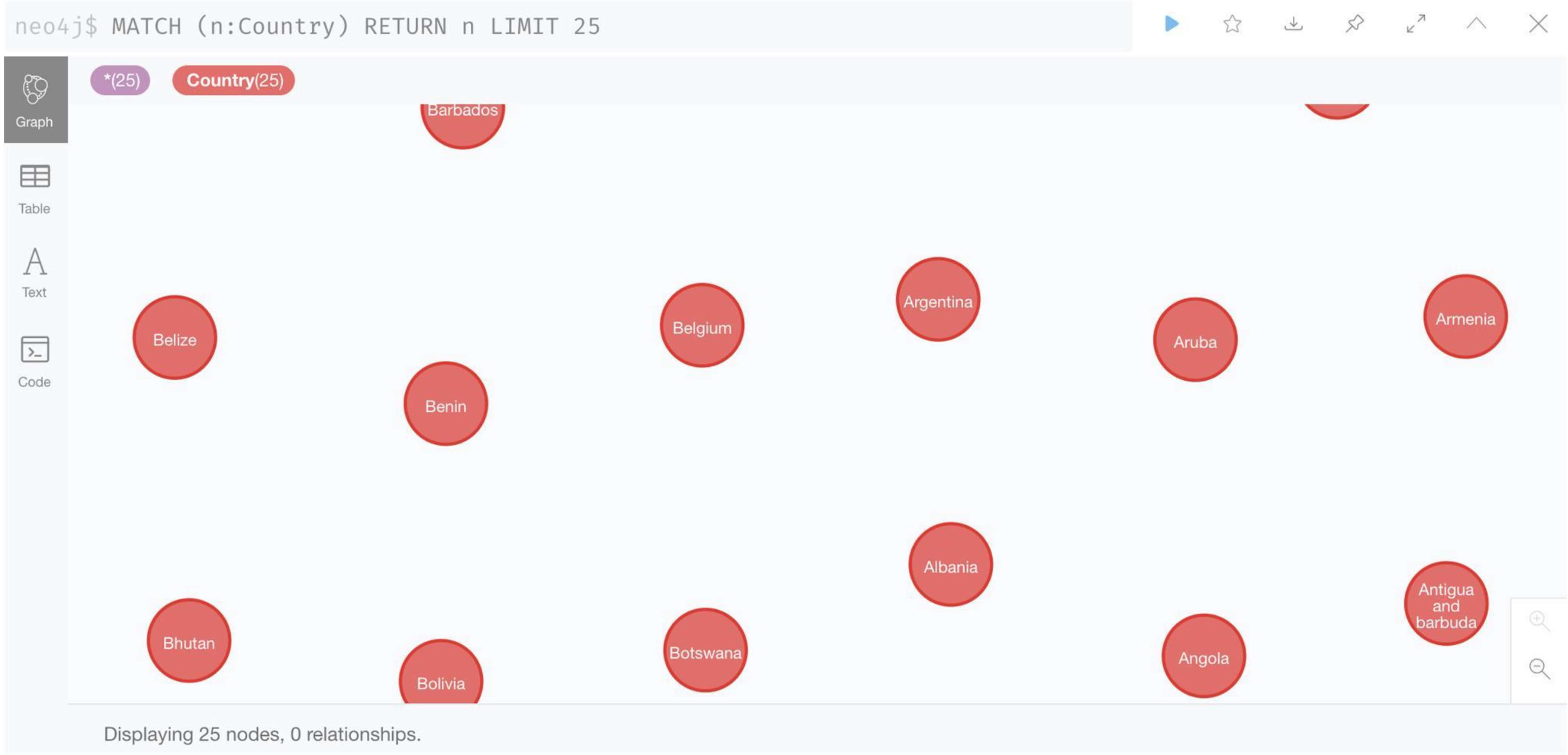


```
neo4j$ // The RelationShip between Country and Province :auto USING PERIODIC CO…
```

```
Created 966 relationships, completed after 7851 ms.
```

Created 966 relationships, completed after 7851 ms.

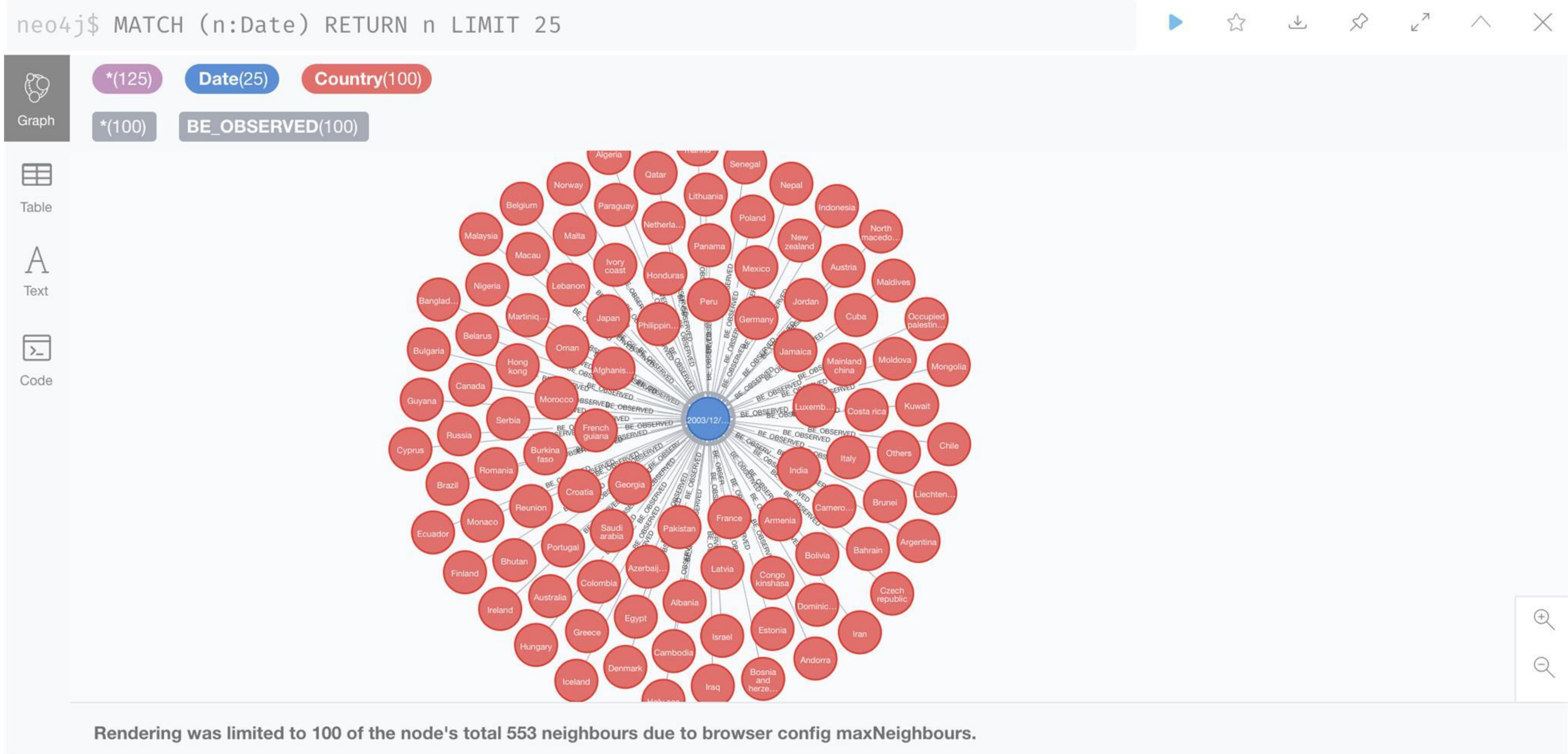
NEO4J Screenshot - Nodes



```
neo4j$ MATCH (n:Country) RETURN n LIMIT 25
```

Displaying 25 nodes, 0 relationships.

NEO4J Screenshot - Relationships



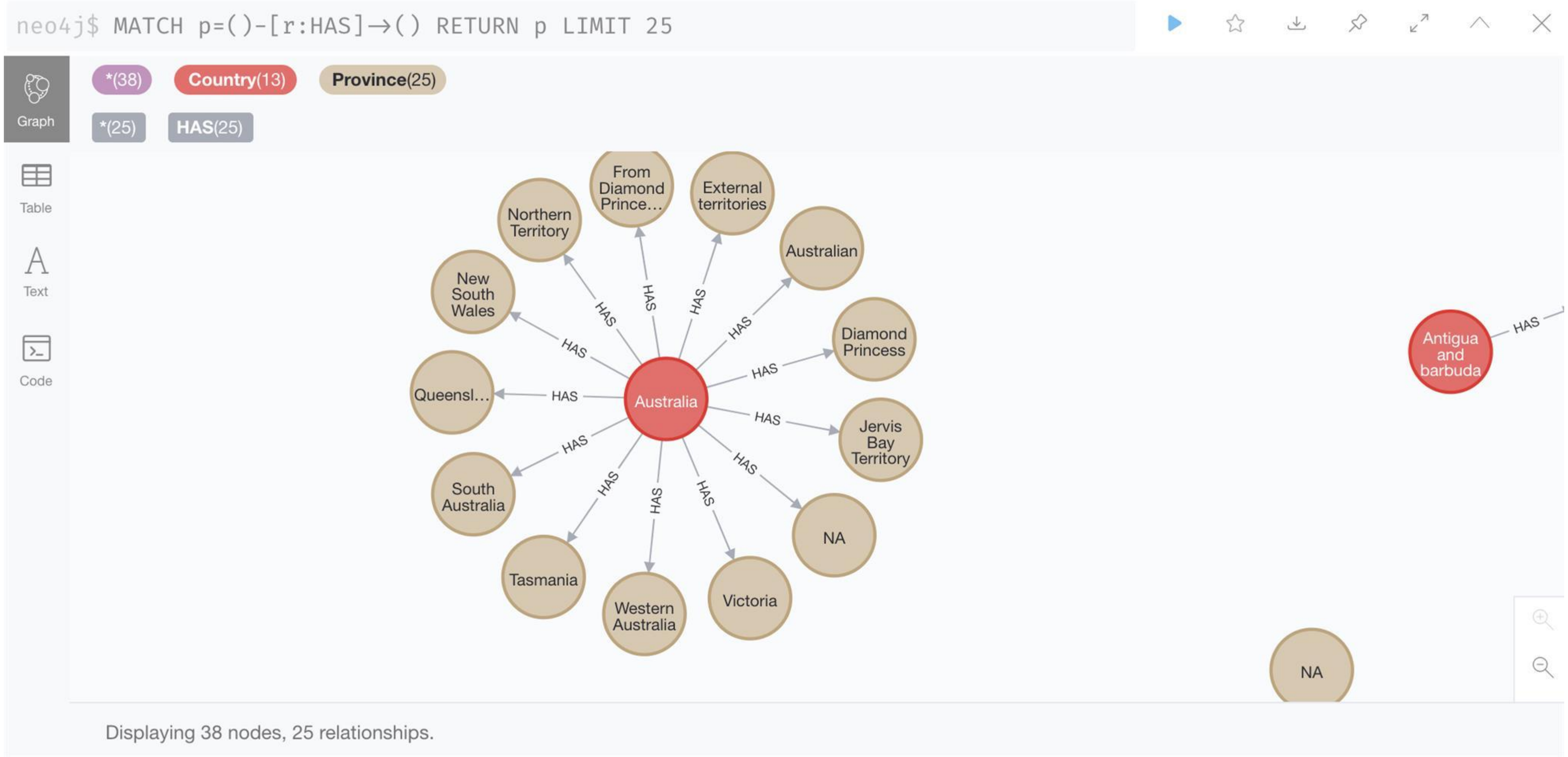
```
neo4j$ MATCH (n:Date) RETURN n LIMIT 25
```

*(125) Date(25) Country(100)

*(100) BE_OBSERVED(100)

Rendering was limited to 100 of the node's total 553 neighbours due to browser config maxNeighbours.

NEO4J Screenshot - Relationships



```
neo4j$ MATCH p=()-[r:HAS]→() RETURN p LIMIT 25
```

Displaying 38 nodes, 25 relationships.

TECHNICAL METADATA

Technical Metadata

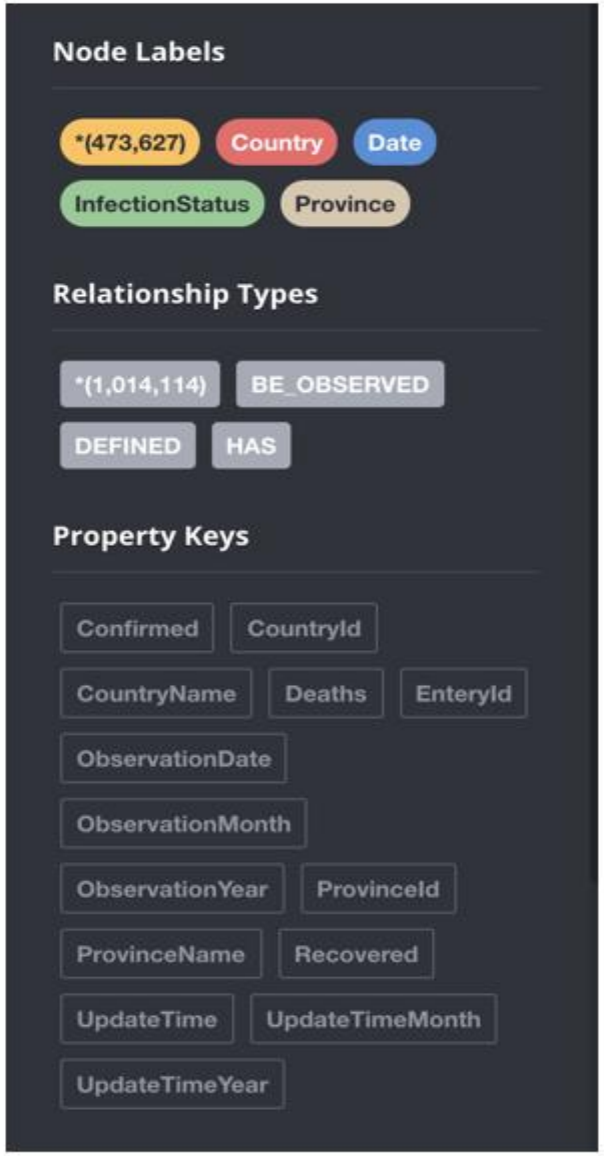
Basic System Requirements

File Name	Covid19.csv
File Size	Covid.19: 16 MB,
Date /Time Created	March 21,2021
Type of Compression	Zip
OS used to run software	Windows
Hardware Processor Name	Intel(R) Core i7
Hardware RAM	16 GB
Tools Used	PowerBI, Anaconda, Microsoft Excel, Smart Draw, Velero ETP, Neo4j

Python Data Profiling, Cleaning and Wrangling

Columns changed or created	CountryId, ProvinceID Observation
Data Types changed: **CountryId**----------------------------→ **ProvinceId**---------------------------→	Decimal Number to String Decimal Number to String
Data Values changes	df1["ProvinceName"] = df1.groupby(['CountryName']) ['ProvinceName']. transform(lambda x: x.fillna)

Neo4j



Node Labels

*(473,627) Country Date InfectionStatus Province

Relationship Types

*(1,014,114) BE_OBSERVED DEFINED HAS

Property Keys

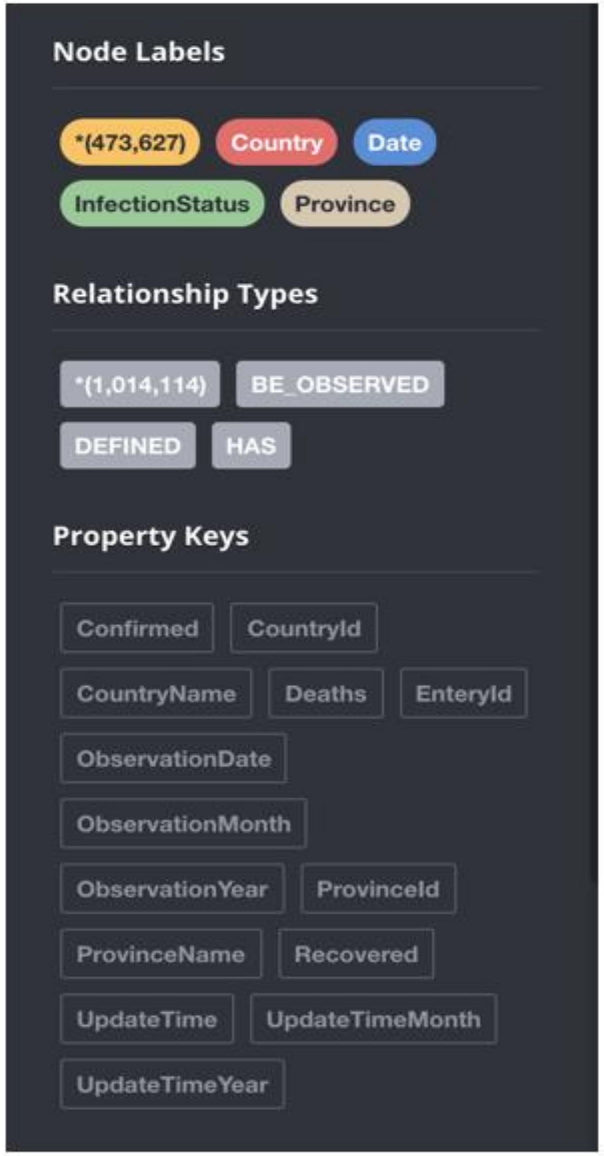
Confirmed CountryId CountryName Deaths EnteryId ObservationDate ObservationMonth ObservationYear ProvinceId ProvinceName Recovered UpdateTime UpdateTimeMonth UpdateTimeYear

Technical Metadata

Covid19.csv

Property	Type
CountryId	STRING
CountryName	STRING
ProvinceName	STRING
ProvinceId	STRING
ObservationDate	DATE_TIME
UpdateTime	DATE_TIME
Recovered	INTEGER
EnteryId	STRING
Deaths	INTEGER
Confirmed	INTEGER
ObservationYear	INTEGER
ObservationMonth	INTEGER
UpdatedTimeYear	INTEGER
UpdateTimeMonth	INTEGER

Neo4j



Node Labels

*(473,627) Country Date InfectionStatus Province

Relationship Types

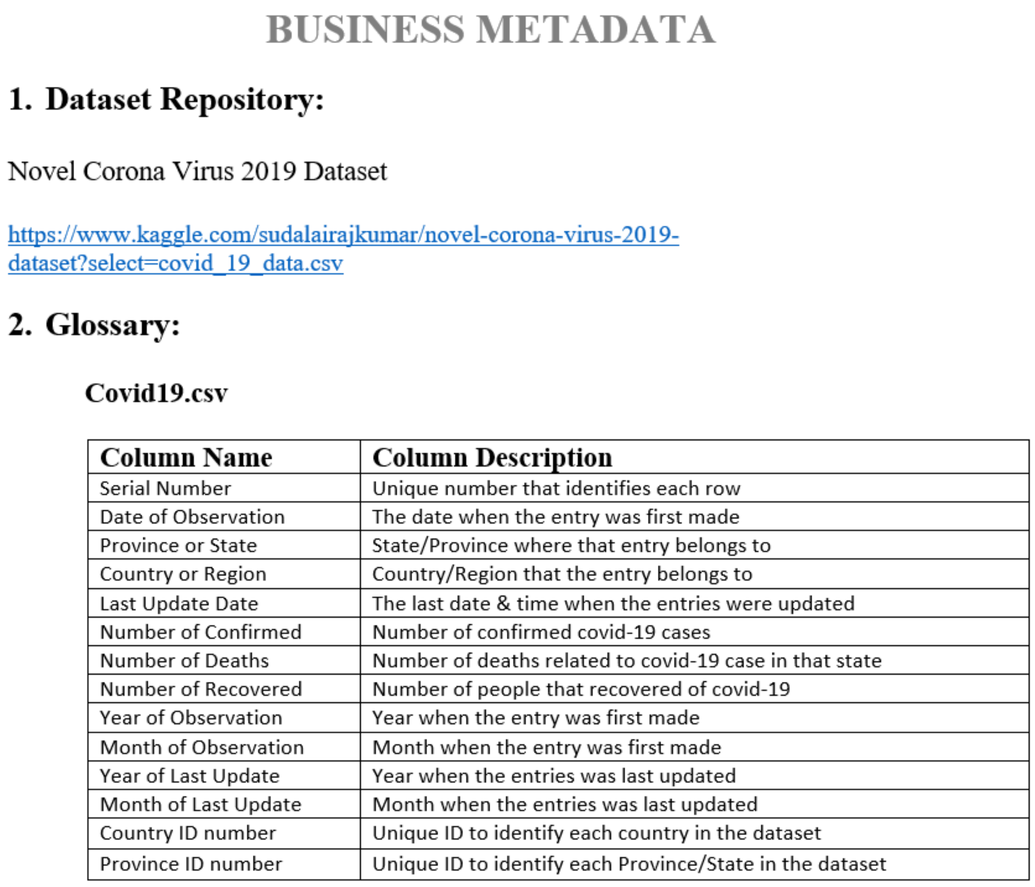
*(1,014,114) BE_OBSERVED DEFINED HAS

Property Keys

Confirmed CountryId CountryName Deaths EnteryId ObservationDate ObservationMonth ObservationYear ProvinceId ProvinceName Recovered UpdateTime UpdateTimeMonth UpdateTimeYear

BUSINESS METADATA

Business Metadata



BUSINESS METADATA

1. Dataset Repository:

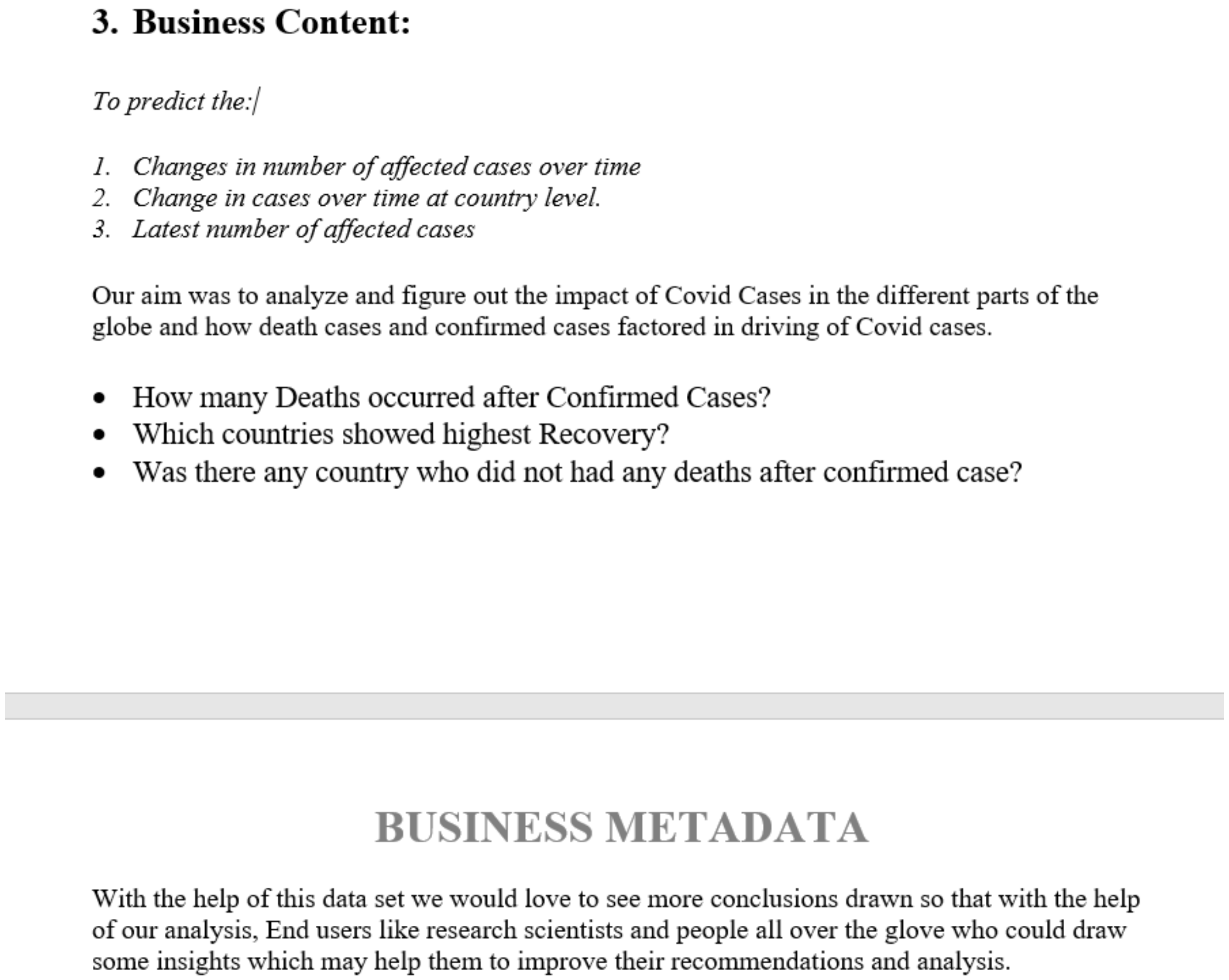
Novel Corona Virus 2019 Dataset

https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset?select=covid_19_data.csv

2. Glossary:

Covid19.csv

Column Name	Column Description
Serial Number	Unique number that identifies each row
Date of Observation	The date when the entry was first made
Province or State	State/Province where that entry belongs to
Country or Region	Country/Region that the entry belongs to
Last Update Date	The last date & time when the entries were updated
Number of Confirmed	Number of confirmed covid-19 cases
Number of Deaths	Number of deaths related to covid-19 case in that state
Number of Recovered	Number of people that recovered of covid-19
Year of Observation	Year when the entry was first made
Month of Observation	Month when the entry was first made
Year of Last Update	Year when the entries was last updated
Month of Last Update	Month when the entries was last updated
Country ID number	Unique ID to identify each country in the dataset
Province ID number	Unique ID to identify each Province/State in the dataset



3. Business Content:

To predict the:

1. *Changes in number of affected cases over time*
2. *Change in cases over time at country level.*
3. *Latest number of affected cases*

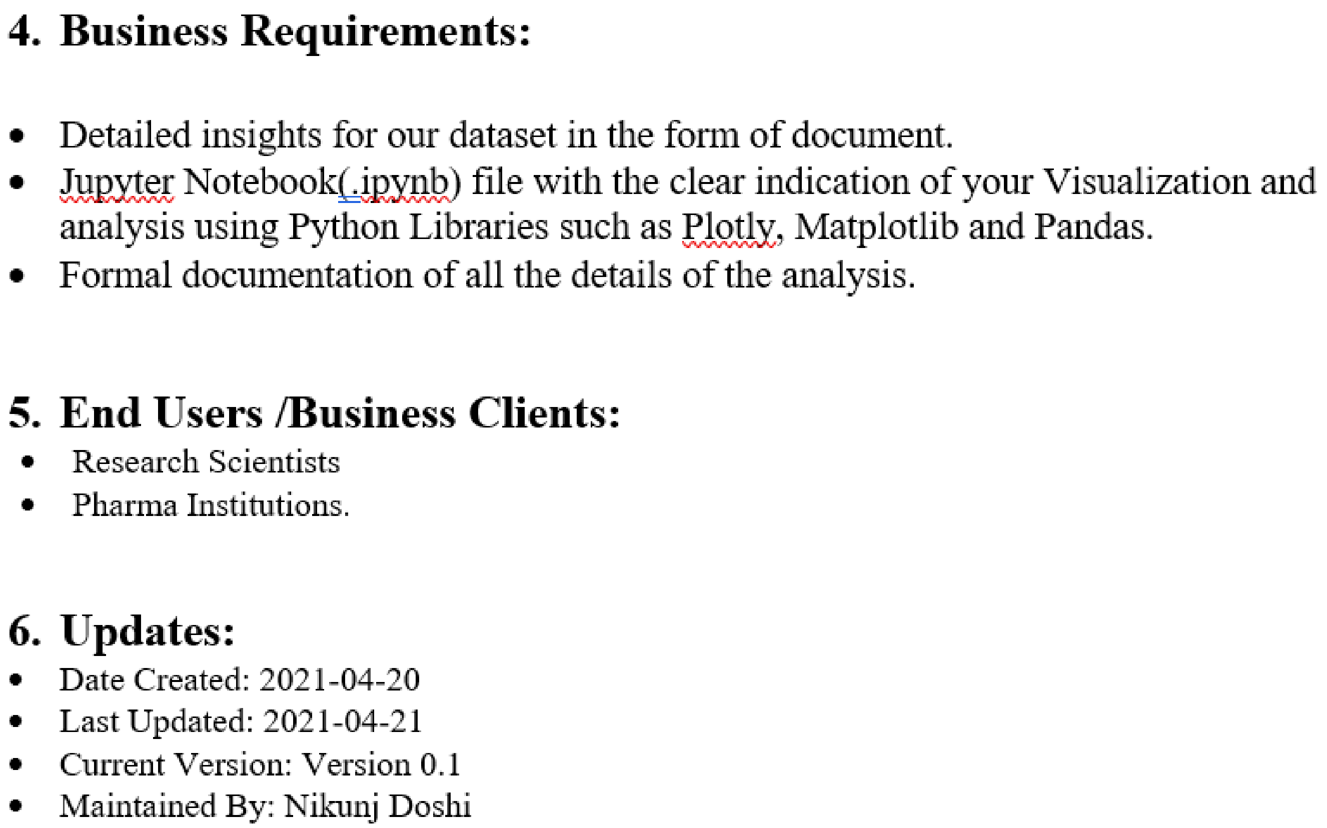
Our aim was to analyze and figure out the impact of Covid Cases in the different parts of the globe and how death cases and confirmed cases factored in driving of Covid cases.

- How many Deaths occurred after Confirmed Cases?
- Which countries showed highest Recovery?
- Was there any country who did not had any deaths after confirmed case?

BUSINESS METADATA

With the help of this data set we would love to see more conclusions drawn so that with the help of our analysis, End users like research scientists and people all over the glove who could draw some insights which may help them to improve their recommendations and analysis.

Business Metadata



4. Business Requirements:

- Detailed insights for our dataset in the form of document.
- Jupyter Notebook(.ipynb) file with the clear indication of your Visualization and analysis using Python Libraries such as Plotly, Matplotlib and Pandas.
- Formal documentation of all the details of the analysis.

5. End Users /Business Clients:

- Research Scientists
- Pharma Institutions.

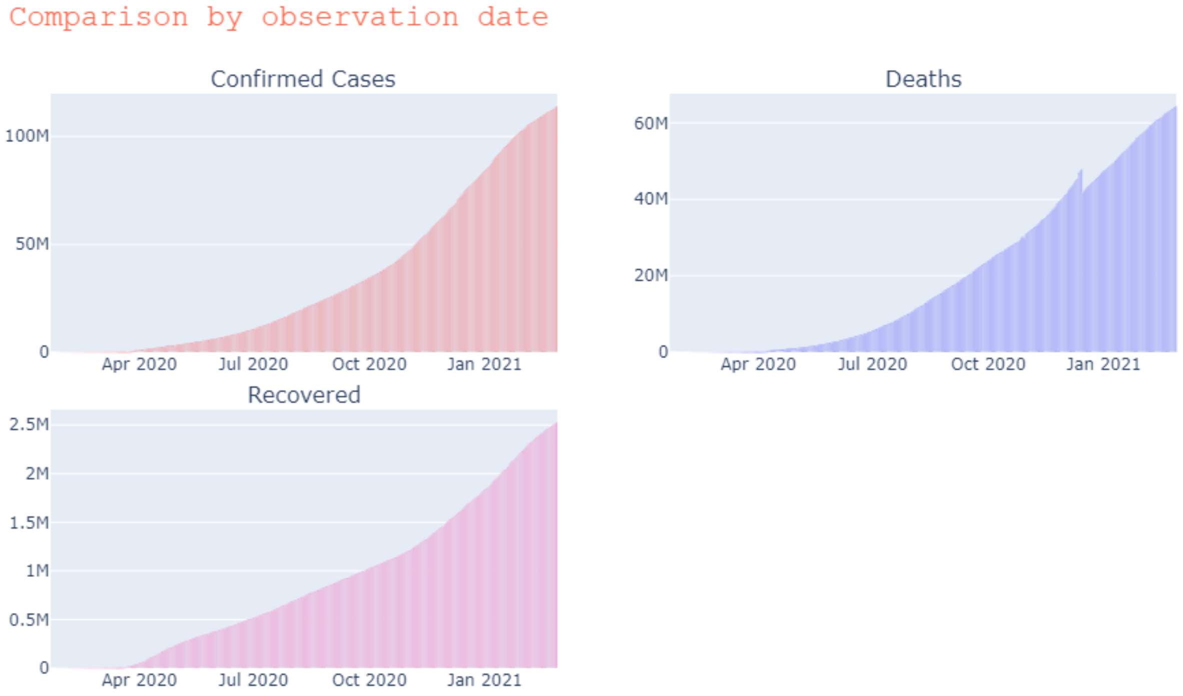
6. Updates:

- Date Created: 2021-04-20
- Last Updated: 2021-04-21
- Current Version: Version 0.1
- Maintained By: Nikunj Doshi

Data Visualizations - Python

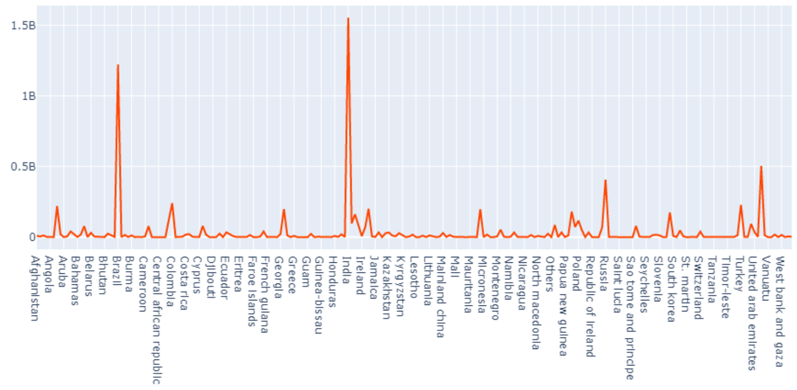
Northeastern University
College of Engineering

- Data Visualization is the graphic representation of data that helps understand the data without requiring technical knowledge
- Data validation & Data visualization were performed using Python
- What was the difference between the confirmed, recovered and death numbers for different month & year?
- What was the country that had highest number of deaths?
- What was the country that had highest number of recovered patients?
- What was the country that had highest number of confirmed cases?
- Which State had highest number of deaths?
- What was the difference between the confirmed, recovered and death numbers for different countries?
- What was the difference in number of deaths in year 2020 and year 2021?
- What was the difference in number of patients that recovered from covid in the year 2020 and year 2021?

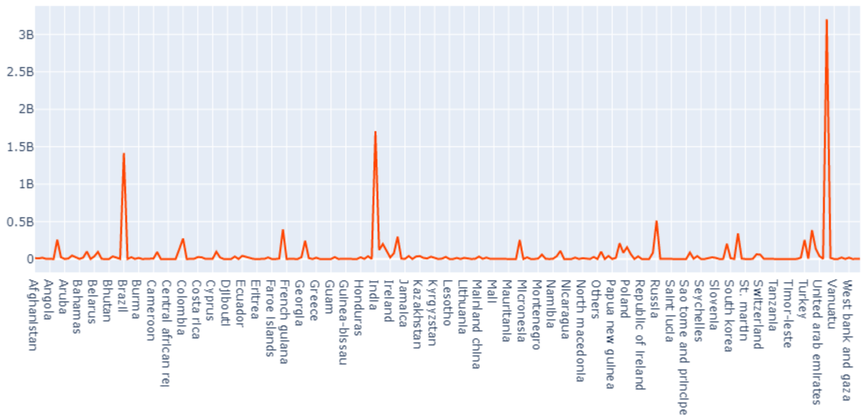


Visualizations

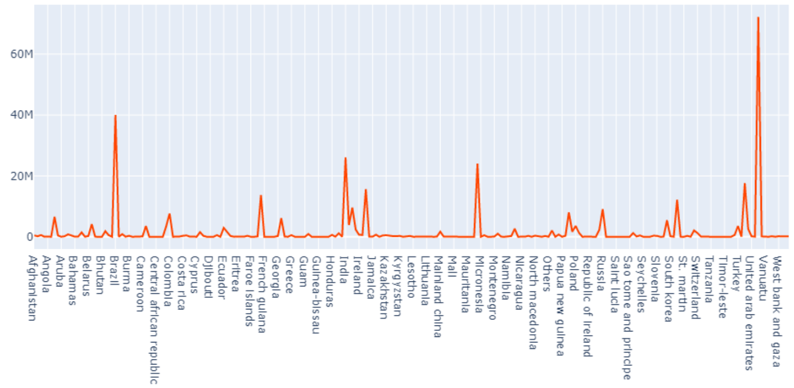
Recovered by Country



Confirmed cases by Country



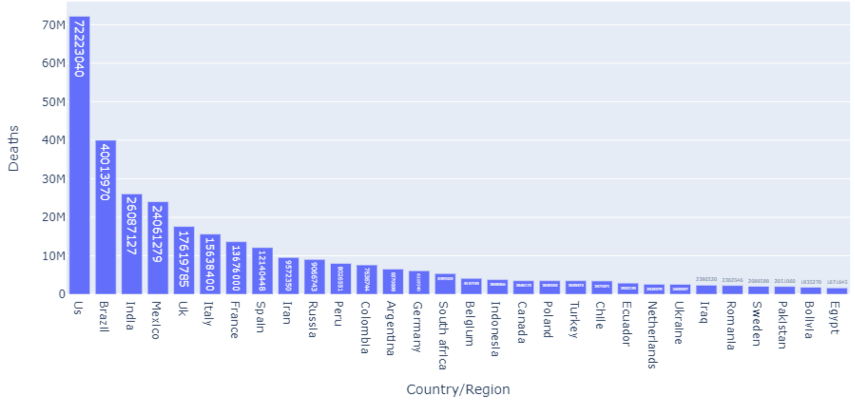
Deaths by Country



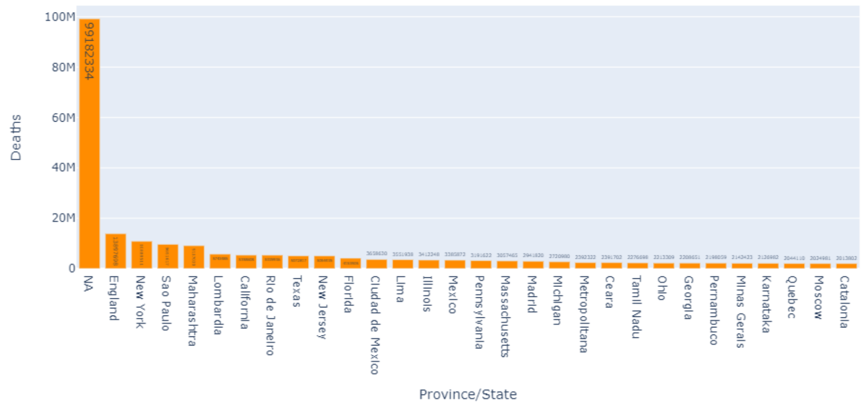
Visualizations

Northeastern University
College of Engineering

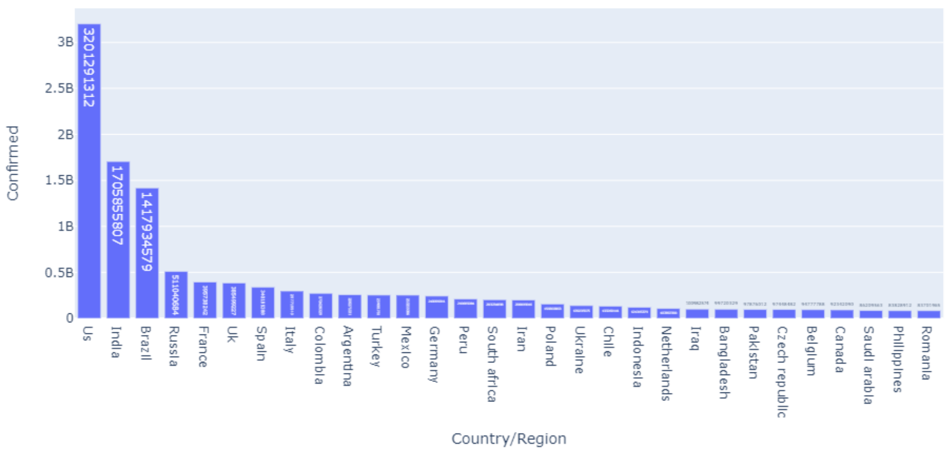
Top 30 countries with highest number of deaths



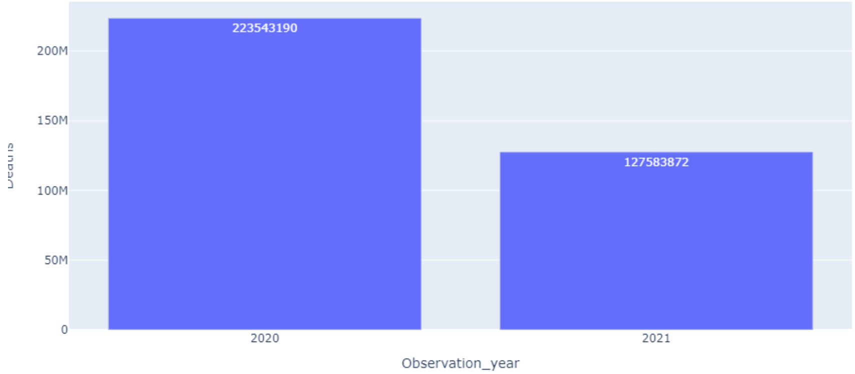
Top 30 States with highest number of deaths



Top 30 countries with highest confirmed cases



Comparison of deaths in year 2020 & 2021

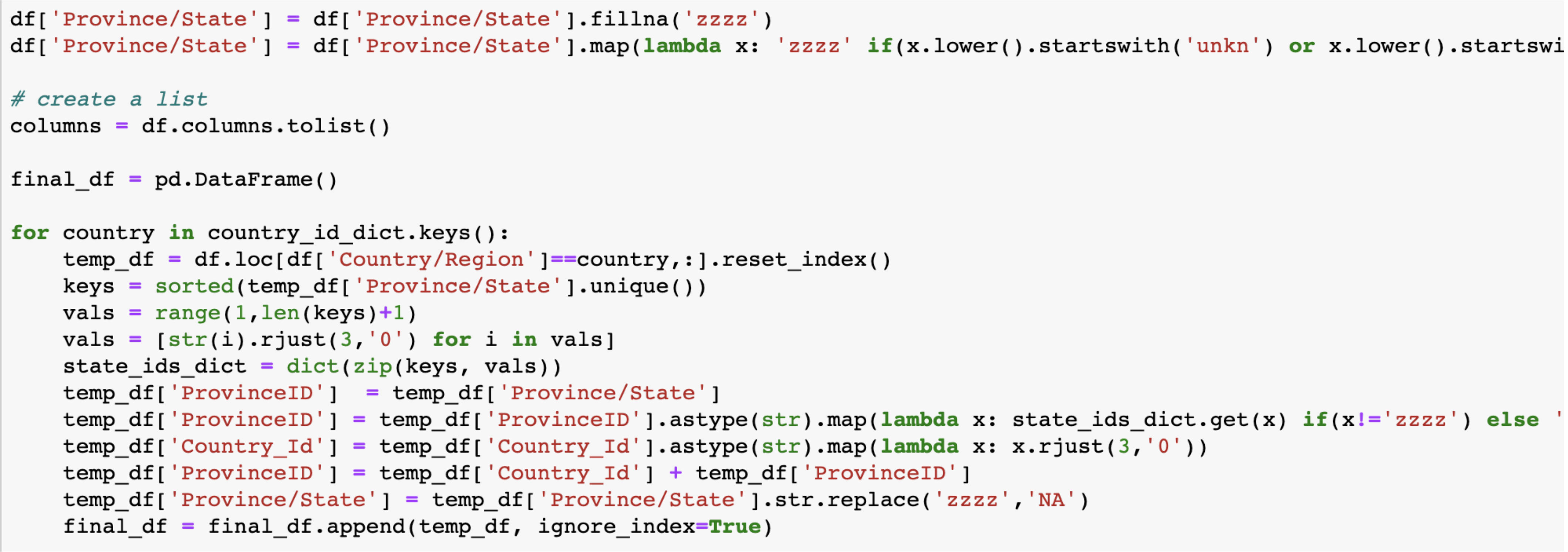


CHALLENGES

Challenges

Challenge 1: To generate the unique Province Id

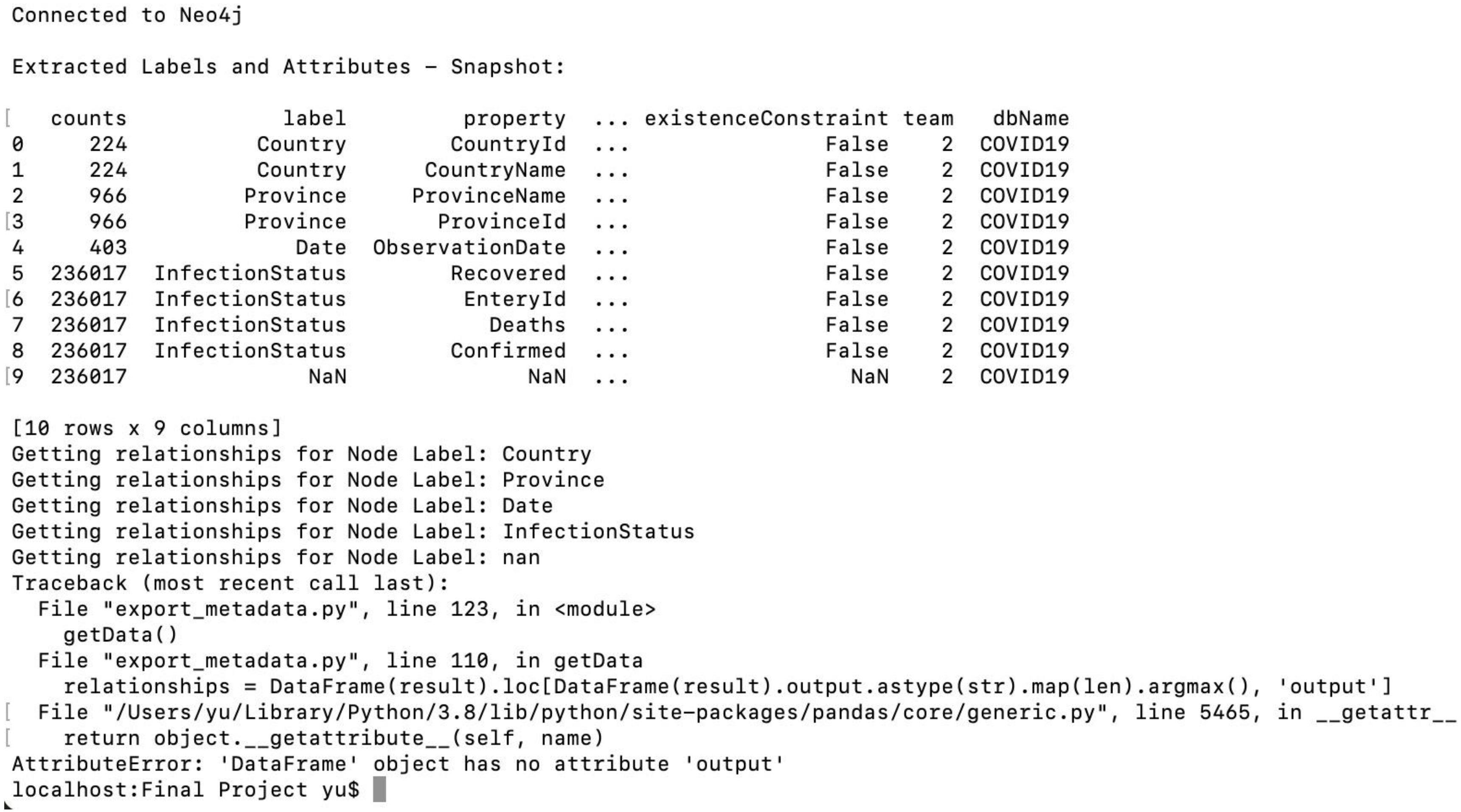
Resolution 1:



```
df['Province/State'] = df['Province/State'].fillna('zzzz')
df['Province/State'] = df['Province/State'].map(lambda x: 'zzzz' if(x.lower().startswith('unkn') or x.lower().startswi

# create a list
columns = df.columns.tolist()

final_df = pd.DataFrame()

for country in country_id_dict.keys():
    temp_df = df.loc[df['Country/Region']==country,:].reset_index()
    keys = sorted(temp_df['Province/State'].unique())
    vals = range(1,len(keys)+1)
    vals = [str(i).rjust(3,'0') for i in vals]
    state_ids_dict = dict(zip(keys, vals))
    temp_df['ProvinceID']  = temp_df['Province/State']
    temp_df['ProvinceID'] = temp_df['ProvinceID'].astype(str).map(lambda x: state_ids_dict.get(x) if(x!='zzzz') else '
    temp_df['Country_Id'] = temp_df['Country_Id'].astype(str).map(lambda x: x.rjust(3,'0'))
    temp_df['ProvinceID'] = temp_df['Country_Id'] + temp_df['ProvinceID']
    temp_df['Province/State'] = temp_df['Province/State'].str.replace('zzzz','NA')
    final_df = final_df.append(temp_df, ignore_index=True)
```

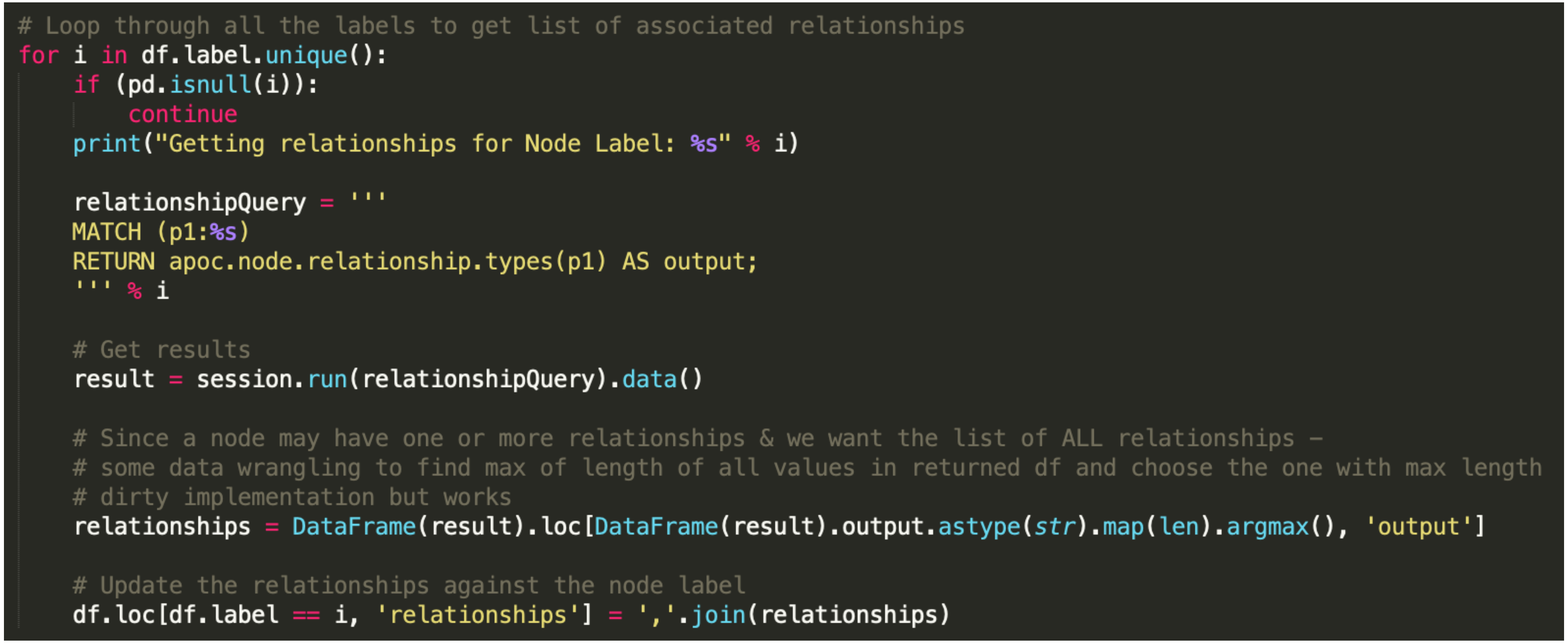
Challenges

Challenge 2: There are some issues when exporting metadata to google excel



```
Connected to Neo4j

Extracted Labels and Attributes - Snapshot:

[   counts           label         property  ... existenceConstraint team   dbName
0      224         Country        CountryId  ...               False    2  COVID19
1      224         Country      CountryName  ...               False    2  COVID19
2      966        Province     ProvinceName  ...               False    2  COVID19
[3     966        Province       ProvinceId  ...               False    2  COVID19
4      403            Date  ObservationDate  ...               False    2  COVID19
5   236017  InfectionStatus       Recovered  ...               False    2  COVID19
[6  236017  InfectionStatus        EnteryId  ...               False    2  COVID19
7   236017  InfectionStatus          Deaths  ...               False    2  COVID19
8   236017  InfectionStatus       Confirmed  ...               False    2  COVID19
[9  236017             NaN              NaN  ...                 NaN    2  COVID19

[10 rows x 9 columns]
Getting relationships for Node Label: Country
Getting relationships for Node Label: Province
Getting relationships for Node Label: Date
Getting relationships for Node Label: InfectionStatus
Getting relationships for Node Label: nan
Traceback (most recent call last):
  File "export_metadata.py", line 123, in <module>
    getData()
  File "export_metadata.py", line 110, in getData
    relationships = DataFrame(result).loc[DataFrame(result).output.astype(str).map(len).argmax(), 'output']
[ File "/Users/yu/Library/Python/3.8/lib/python/site-packages/pandas/core/generic.py", line 5465, in __getattr__
[   return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'output'
localhost:Final Project yu$ 
```

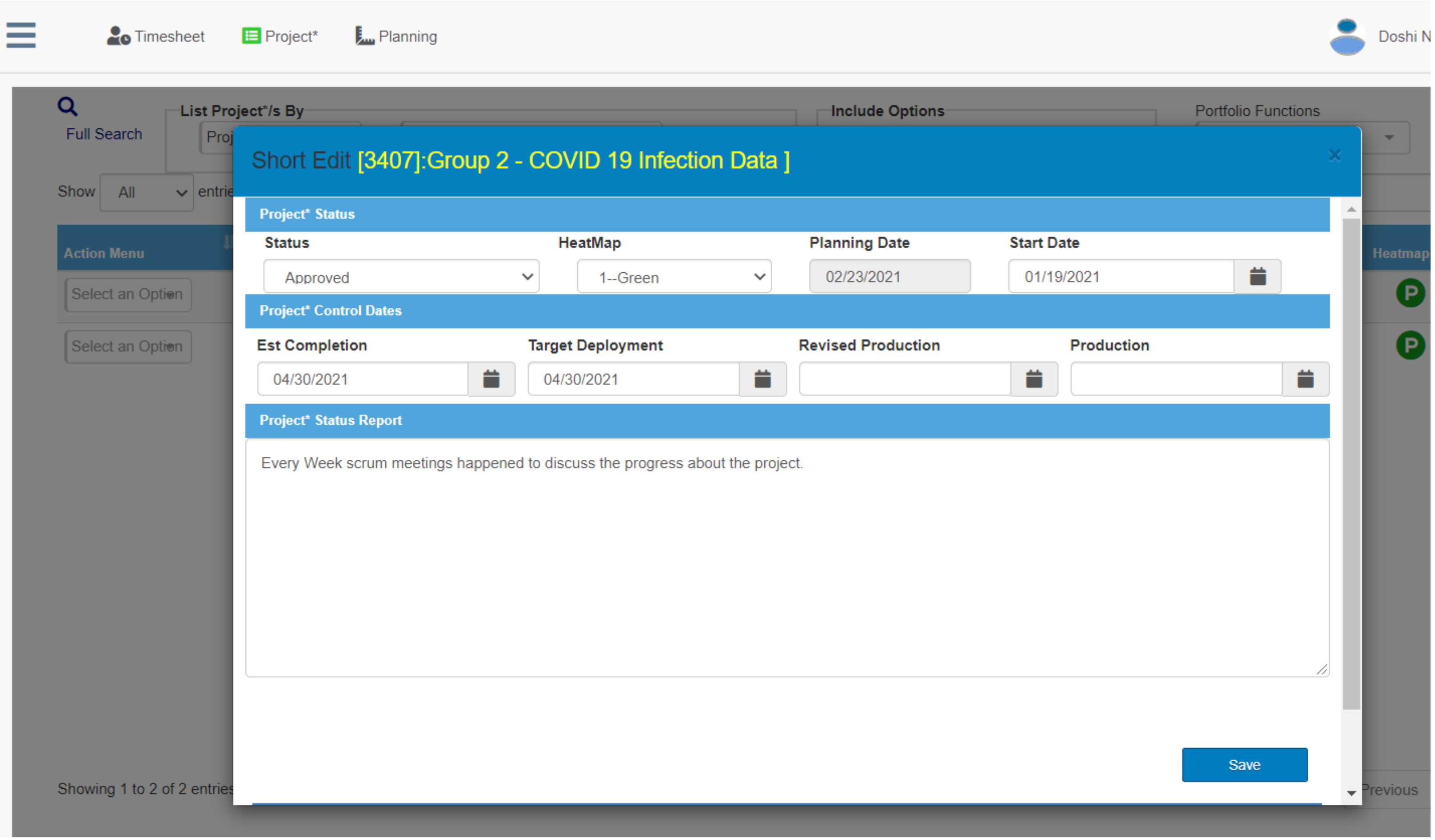
Resolution 2: Filter the abnormal data:



```
# Loop through all the labels to get list of associated relationships
for i in df.label.unique():
    if (pd.isnull(i)):
        continue
    print("Getting relationships for Node Label: %s" % i)

    relationshipQuery = '''
    MATCH (p1:%s)
    RETURN apoc.node.relationship.types(p1) AS output;
    ''' % i

    # Get results
    result = session.run(relationshipQuery).data()

    # Since a node may have one or more relationships & we want the list of ALL relationships -
    # some data wrangling to find max of length of all values in returned df and choose the one with max length
    # dirty implementation but works
    relationships = DataFrame(result).loc[DataFrame(result).output.astype(str).map(len).argmax(), 'output']

    # Update the relationships against the node label
    df.loc[df.label == i, 'relationships'] = ','.join(relationships)
```

VELERO SCREENSHOTS

Velero Screenshot - Short Form



Short Edit [3407]:Group 2 - COVID 19 Infection Data]

Project* Status

Status	HeatMap	Planning Date	Start Date
Approved	1--Green	02/23/2021	01/19/2021

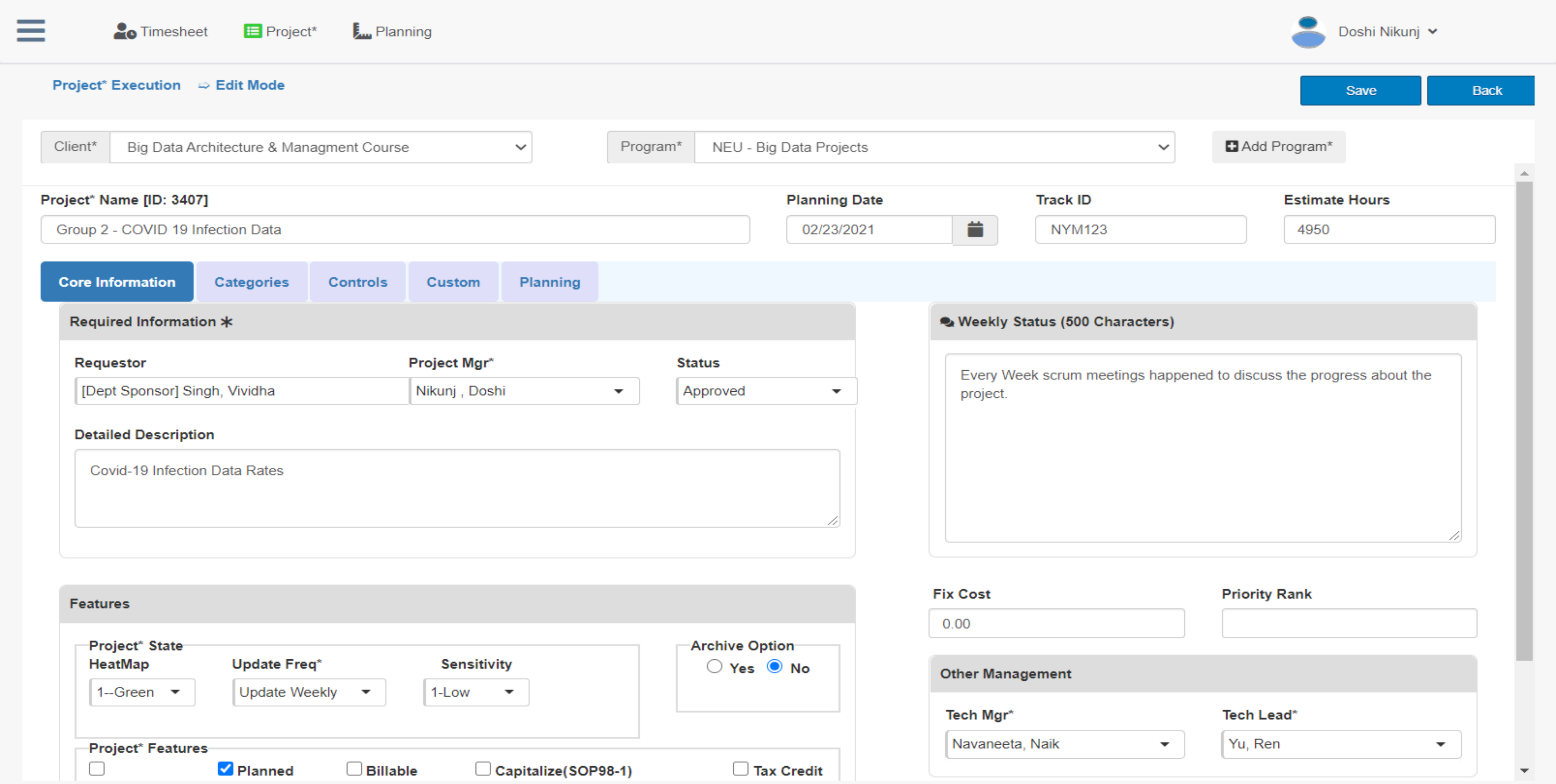
Project* Control Dates

Est Completion	Target Deployment	Revised Production	Production
04/30/2021	04/30/2021		

Project* Status Report

Every Week scrum meetings happened to discuss the progress about the project.

Velero Screenshot – Long Form



Client*: Big Data Architecture & Managment Course

Program*: NEU - Big Data Projects

Project* Name [ID: 3407]: Group 2 - COVID 19 Infection Data

Planning Date: 02/23/2021

Track ID: NYM123

Estimate Hours: 4950

Required Information ∗

Requestor: [Dept Sponsor] Singh, Vividha

Project Mgr*: Nikunj , Doshi

Status: Approved

Detailed Description: Covid-19 Infection Data Rates

Weekly Status (500 Characters)

Every Week scrum meetings happened to discuss the progress about the project.

Fix Cost: 0.00

Priority Rank:

Features

Project* State

HeatMap: 1--Green

Update Freq*: Update Weekly

Sensitivity: 1-Low

Archive Option: ○ Yes ● No

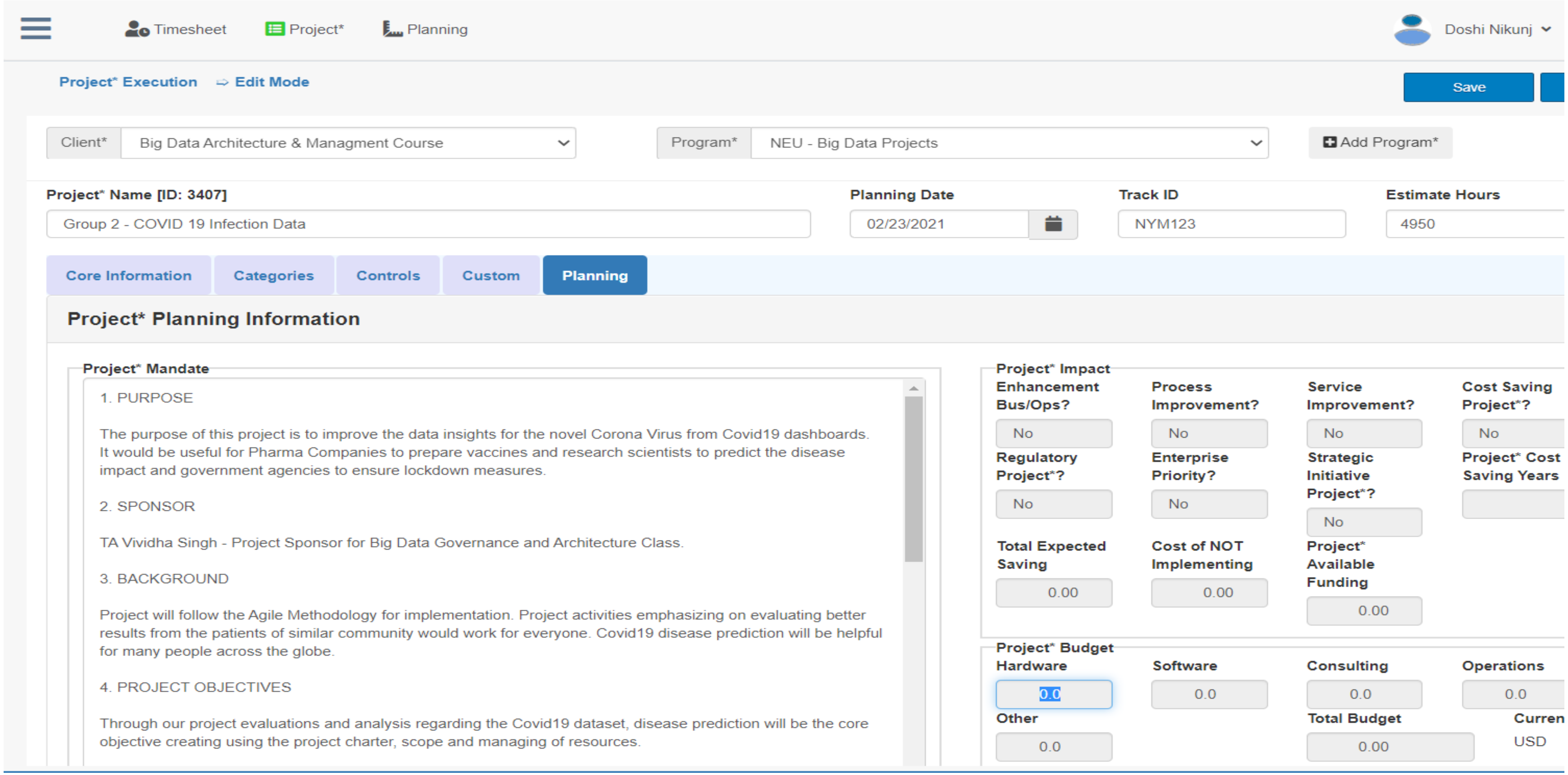
Project* Features: □ ☑ Planned □ Billable □ Capitalize(SOP98-1) □ Tax Credit

Other Management

Tech Mgr*: Navaneeta, Naik

Tech Lead*: Yu, Ren

Velero Screenshot - Mandate



Client*: Big Data Architecture & Managment Course

Program*: NEU - Big Data Projects

Project* Name [ID: 3407]	Planning Date	Track ID	Estimate Hours
Group 2 - COVID 19 Infection Data	02/23/2021	NYM123	4950

Project* Planning Information

Project* Mandate

1. PURPOSE

The purpose of this project is to improve the data insights for the novel Corona Virus from Covid19 dashboards. It would be useful for Pharma Companies to prepare vaccines and research scientists to predict the disease impact and government agencies to ensure lockdown measures.

2. SPONSOR

TA Vividha Singh - Project Sponsor for Big Data Governance and Architecture Class.

3. BACKGROUND

Project will follow the Agile Methodology for implementation. Project activities emphasizing on evaluating better results from the patients of similar community would work for everyone. Covid19 disease prediction will be helpful for many people across the globe.

4. PROJECT OBJECTIVES

Through our project evaluations and analysis regarding the Covid19 dataset, disease prediction will be the core objective creating using the project charter, scope and managing of resources.

Project* Impact

Enhancement Bus/Ops?	Process Improvement?	Service Improvement?	Cost Saving Project*?
No	No	No	No

Regulatory Project*?	Enterprise Priority?	Strategic Initiative Project*?	Project* Cost Saving Years
No	No	No	

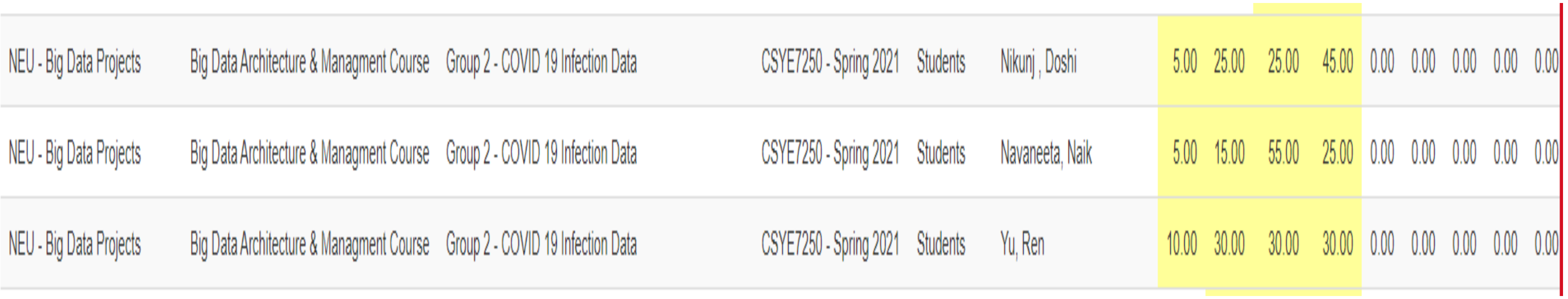
Total Expected Saving	Cost of NOT Implementing	Project* Available Funding
0.00	0.00	0.00

Project* Budget

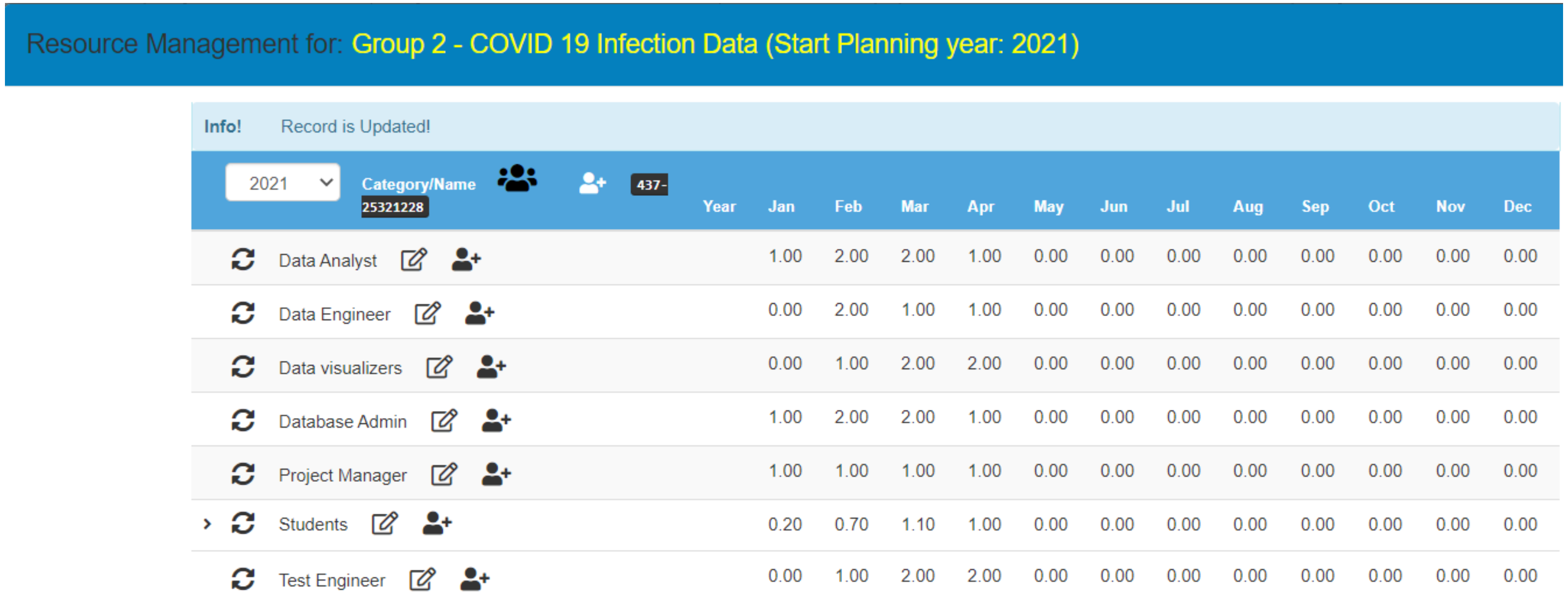
Hardware	Software	Consulting	Operations
0.0	0.0	0.0	0.0

Other	Total Budget	Curren
0.0	0.00	USD

Velero Screenshot – Resource Management



NEU - Big Data Projects	Big Data Architecture & Managment Course	Group 2 - COVID 19 Infection Data	CSYE7250 - Spring 2021	Students	Nikunj , Doshi	5.00	25.00	25.00	45.00	0.00	0.00	0.00	0.00	0.00
NEU - Big Data Projects	Big Data Architecture & Managment Course	Group 2 - COVID 19 Infection Data	CSYE7250 - Spring 2021	Students	Navaneeta, Naik	5.00	15.00	55.00	25.00	0.00	0.00	0.00	0.00	0.00
NEU - Big Data Projects	Big Data Architecture & Managment Course	Group 2 - COVID 19 Infection Data	CSYE7250 - Spring 2021	Students	Yu, Ren	10.00	30.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00

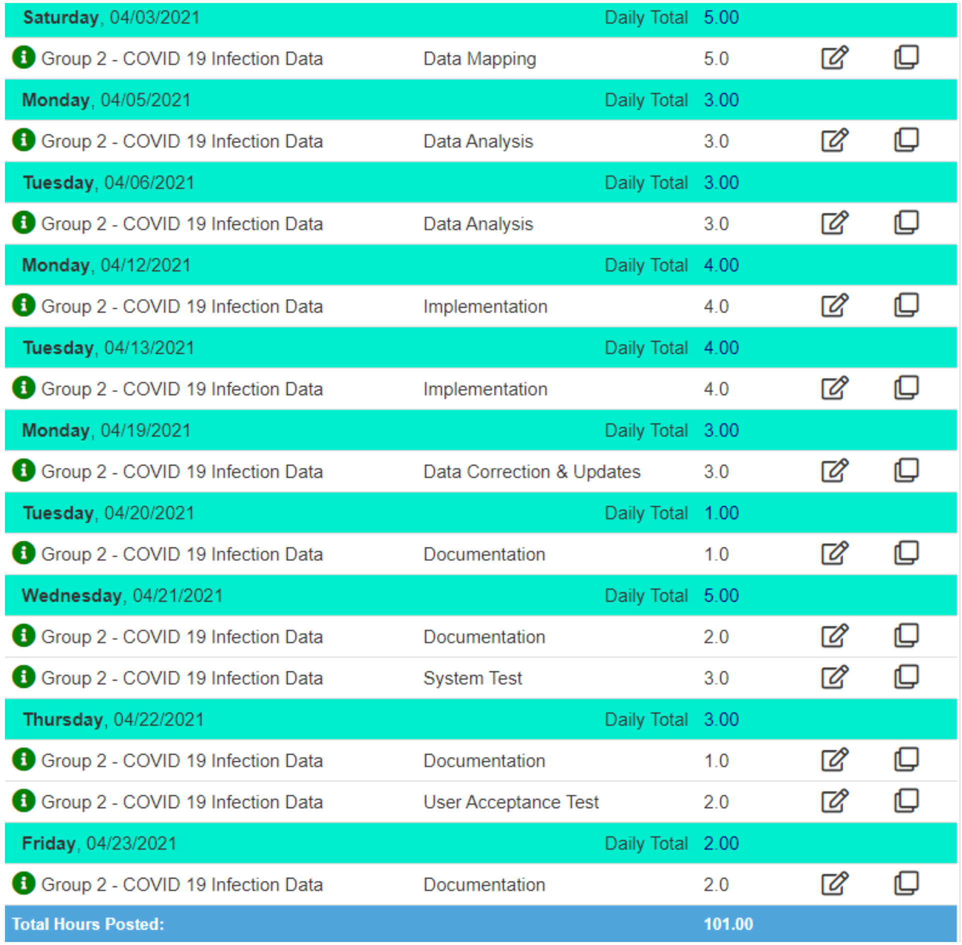


Resource Management for: Group 2 - COVID 19 Infection Data (Start Planning year: 2021)

Info! Record is Updated!

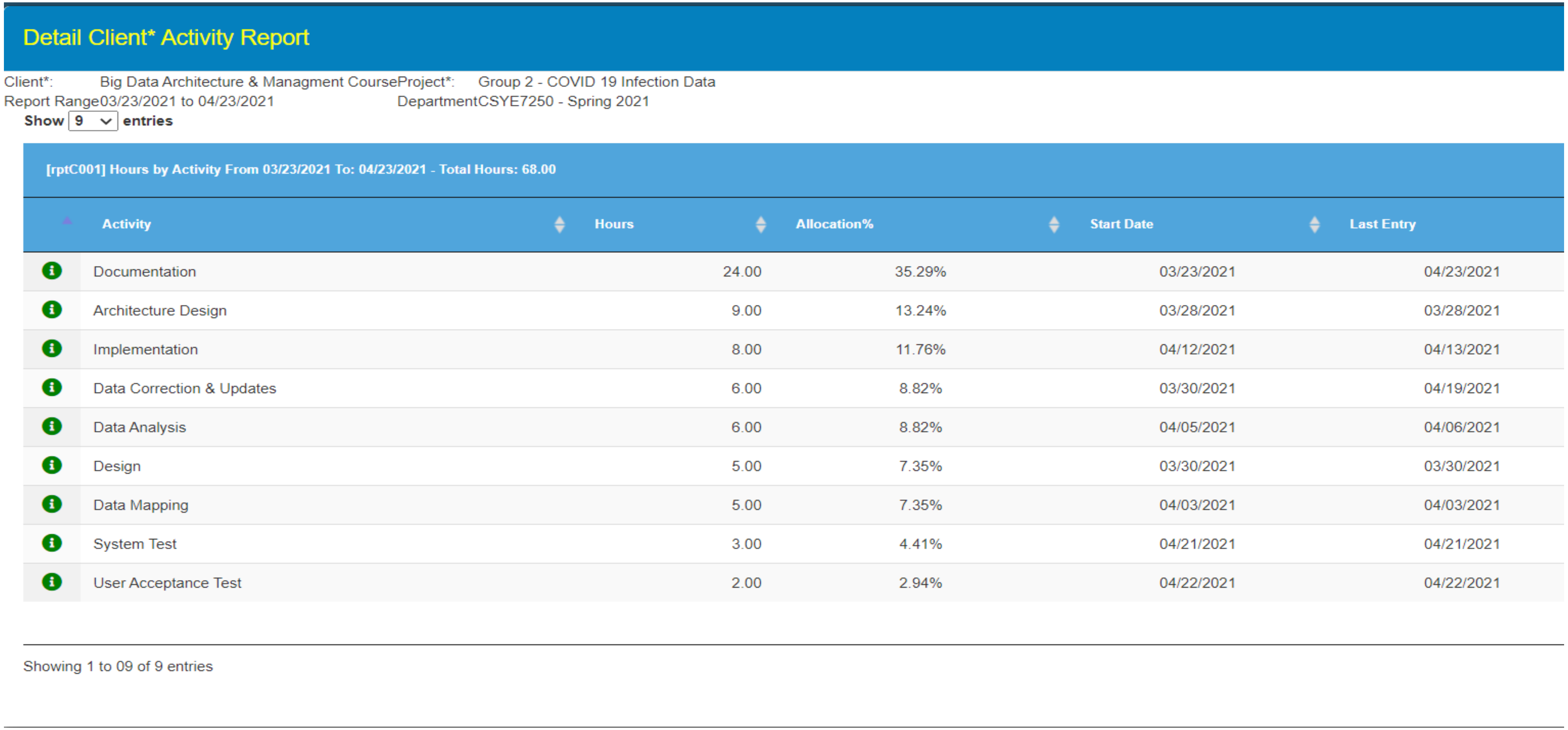
2021	Category/Name 25321228	437-	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	Data Analyst			1.00	2.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Data Engineer			0.00	2.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Data visualizers			0.00	1.00	2.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Database Admin			1.00	2.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Project Manager			1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
›	Students			0.20	0.70	1.10	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Test Engineer			0.00	1.00	2.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Velero Screenshot – Time Sheet



Saturday, 04/03/2021		Daily Total	5.00
Group 2 - COVID 19 Infection Data	Data Mapping		5.0
Monday, 04/05/2021		Daily Total	3.00
Group 2 - COVID 19 Infection Data	Data Analysis		3.0
Tuesday, 04/06/2021		Daily Total	3.00
Group 2 - COVID 19 Infection Data	Data Analysis		3.0
Monday, 04/12/2021		Daily Total	4.00
Group 2 - COVID 19 Infection Data	Implementation		4.0
Tuesday, 04/13/2021		Daily Total	4.00
Group 2 - COVID 19 Infection Data	Implementation		4.0
Monday, 04/19/2021		Daily Total	3.00
Group 2 - COVID 19 Infection Data	Data Correction & Updates		3.0
Tuesday, 04/20/2021		Daily Total	1.00
Group 2 - COVID 19 Infection Data	Documentation		1.0
Wednesday, 04/21/2021		Daily Total	5.00
Group 2 - COVID 19 Infection Data	Documentation		2.0
Group 2 - COVID 19 Infection Data	System Test		3.0
Thursday, 04/22/2021		Daily Total	3.00
Group 2 - COVID 19 Infection Data	Documentation		1.0
Group 2 - COVID 19 Infection Data	User Acceptance Test		2.0
Friday, 04/23/2021		Daily Total	2.00
Group 2 - COVID 19 Infection Data	Documentation		2.0
Total Hours Posted:			**101.00**

Velero Screenshot - Activity Allocation



Detail Client* Activity Report

Client*: Big Data Architecture & Managment CourseProject*: Group 2 - COVID 19 Infection Data
Report Range03/23/2021 to 04/23/2021 DepartmentCSYE7250 - Spring 2021

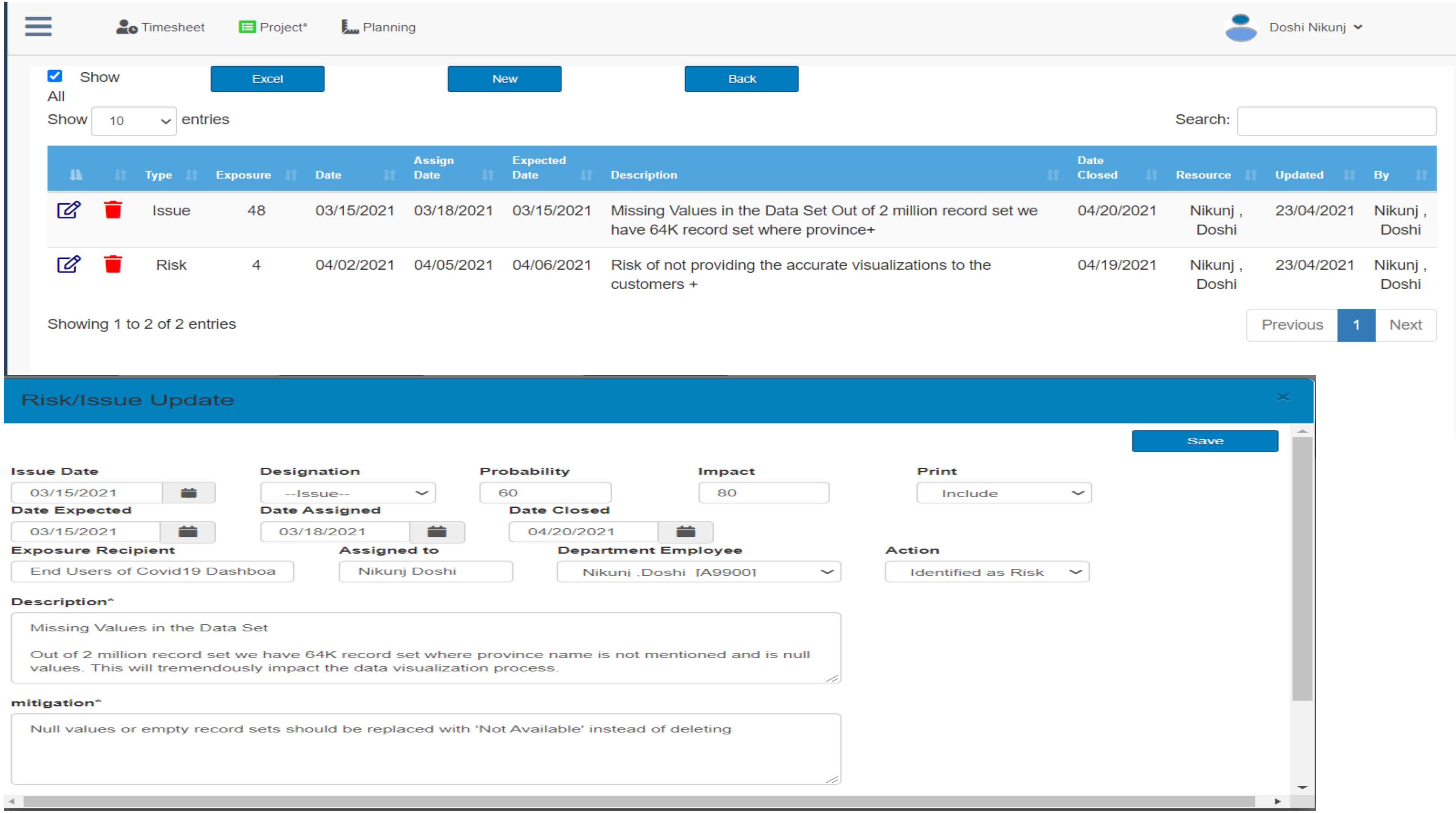
Show 9 **entries**

[rptC001] Hours by Activity From 03/23/2021 To: 04/23/2021 - Total Hours: 68.00

	Activity	Hours	Allocation%	Start Date	Last Entry
ℹ	Documentation	24.00	35.29%	03/23/2021	04/23/2021
ℹ	Architecture Design	9.00	13.24%	03/28/2021	03/28/2021
ℹ	Implementation	8.00	11.76%	04/12/2021	04/13/2021
ℹ	Data Correction & Updates	6.00	8.82%	03/30/2021	04/19/2021
ℹ	Data Analysis	6.00	8.82%	04/05/2021	04/06/2021
ℹ	Design	5.00	7.35%	03/30/2021	03/30/2021
ℹ	Data Mapping	5.00	7.35%	04/03/2021	04/03/2021
ℹ	System Test	3.00	4.41%	04/21/2021	04/21/2021
ℹ	User Acceptance Test	2.00	2.94%	04/22/2021	04/22/2021

Showing 1 to 09 of 9 entries

Velero Screenshot - Risks & Issues



☑ Show All

Show 10 entries

Search:

		Type	Exposure	Date	Assign Date	Expected Date	Description	Date Closed	Resource	Updated	By
		Issue	48	03/15/2021	03/18/2021	03/15/2021	Missing Values in the Data Set Out of 2 million record set we have 64K record set where province+	04/20/2021	Nikunj , Doshi	23/04/2021	Nikunj , Doshi
		Risk	4	04/02/2021	04/05/2021	04/06/2021	Risk of not providing the accurate visualizations to the customers +	04/19/2021	Nikunj , Doshi	23/04/2021	Nikunj , Doshi

Showing 1 to 2 of 2 entries

Risk/Issue Update

Issue Date: 03/15/2021

Designation: --Issue--

Probability: 60

Impact: 80

Print: Include

Date Expected: 03/15/2021

Date Assigned: 03/18/2021

Date Closed: 04/20/2021

Exposure Recipient: End Users of Covid19 Dashboa

Assigned to: Nikunj Doshi

Department Employee: Nikunj .Doshi [A9900]

Action: Identified as Risk

Description*

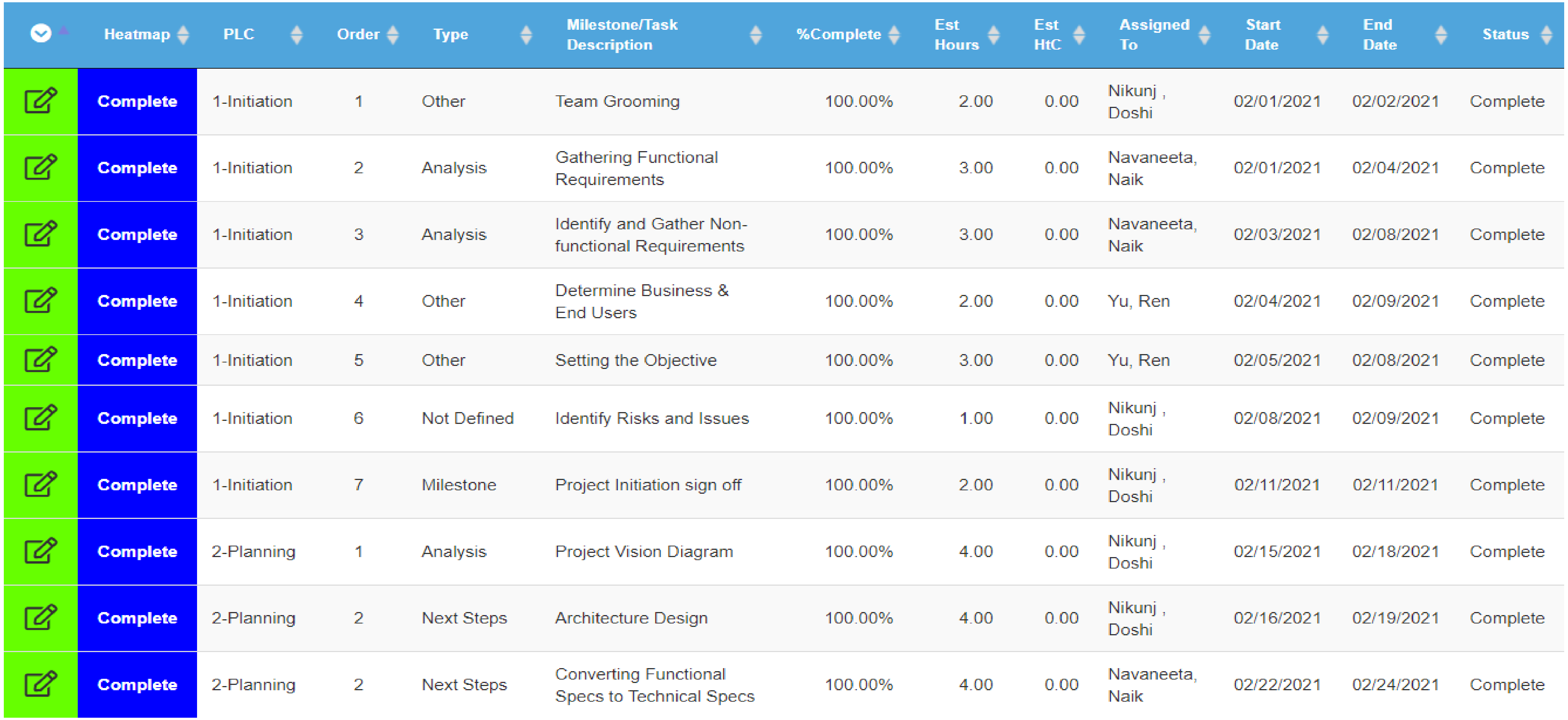
Missing Values in the Data Set

Out of 2 million record set we have 64K record set where province name is not mentioned and is null values. This will tremendously impact the data visualization process.

mitigation*

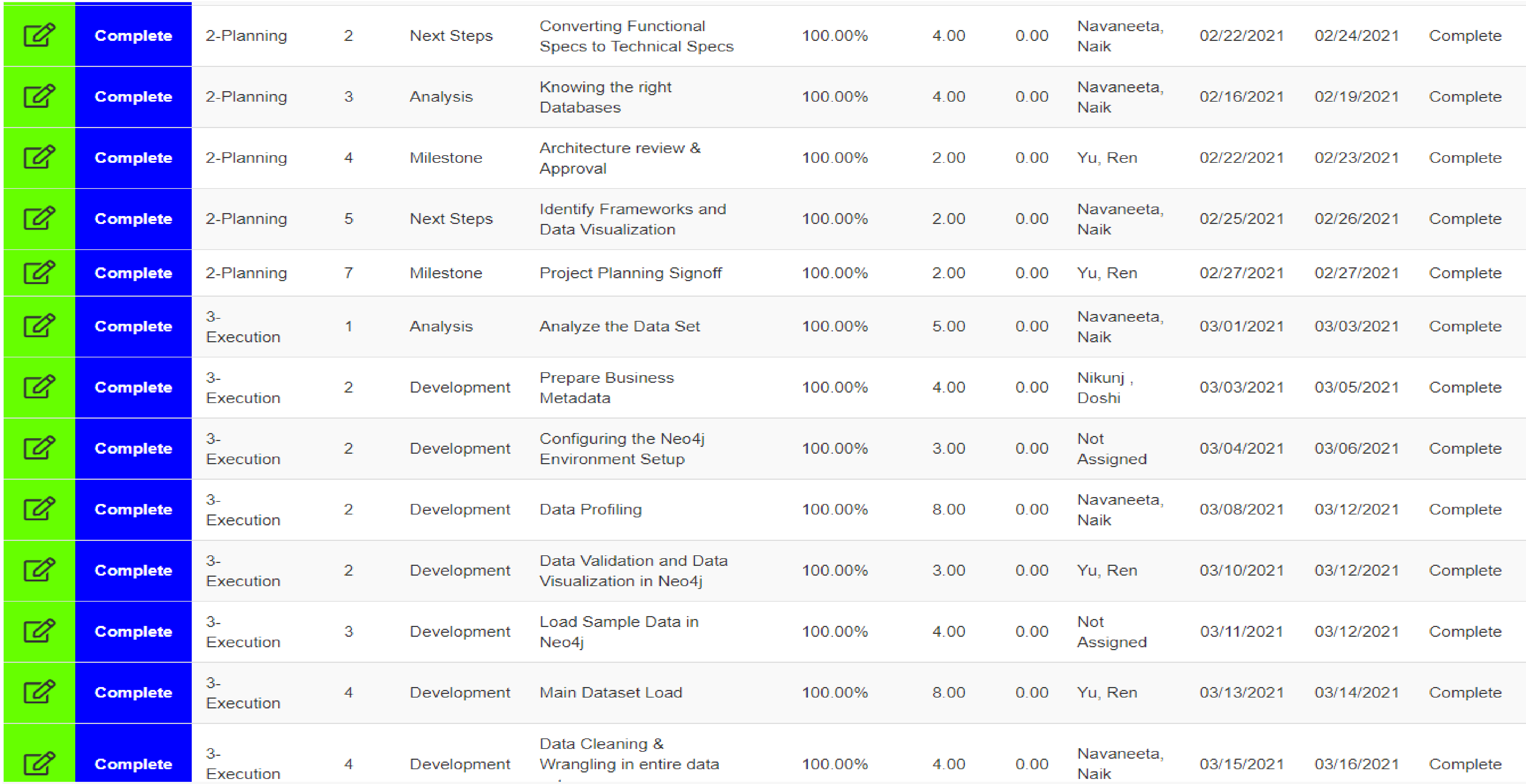
Null values or empty record sets should be replaced with 'Not Available' instead of deleting

Velero Screenshot – Group Allocation 1



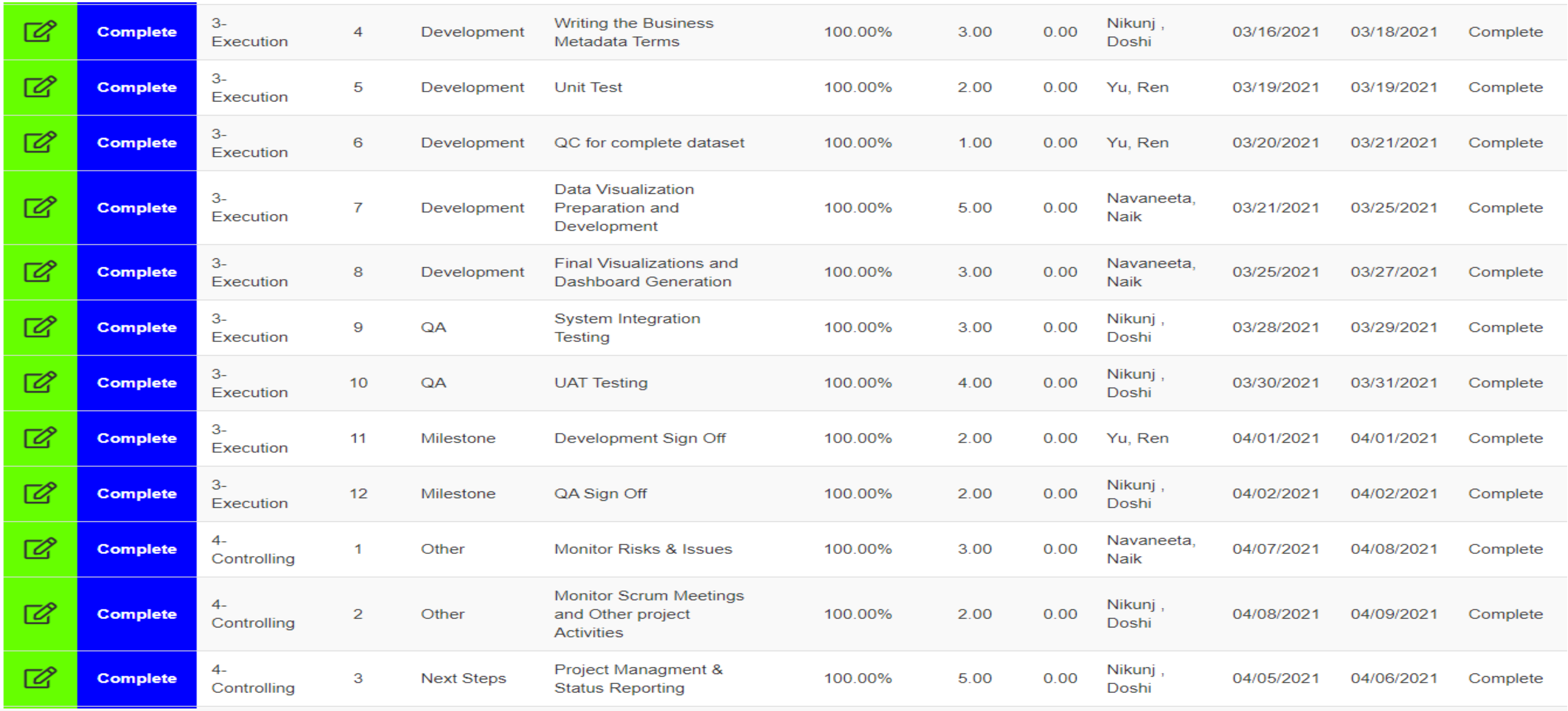
	Heatmap	PLC	Order	Type	Milestone/Task Description	%Complete	Est Hours	Est HtC	Assigned To	Start Date	End Date	Status
✎	Complete	1-Initiation	1	Other	Team Grooming	100.00%	2.00	0.00	Nikunj , Doshi	02/01/2021	02/02/2021	Complete
✎	Complete	1-Initiation	2	Analysis	Gathering Functional Requirements	100.00%	3.00	0.00	Navaneeta, Naik	02/01/2021	02/04/2021	Complete
✎	Complete	1-Initiation	3	Analysis	Identify and Gather Non-functional Requirements	100.00%	3.00	0.00	Navaneeta, Naik	02/03/2021	02/08/2021	Complete
✎	Complete	1-Initiation	4	Other	Determine Business & End Users	100.00%	2.00	0.00	Yu, Ren	02/04/2021	02/09/2021	Complete
✎	Complete	1-Initiation	5	Other	Setting the Objective	100.00%	3.00	0.00	Yu, Ren	02/05/2021	02/08/2021	Complete
✎	Complete	1-Initiation	6	Not Defined	Identify Risks and Issues	100.00%	1.00	0.00	Nikunj , Doshi	02/08/2021	02/09/2021	Complete
✎	Complete	1-Initiation	7	Milestone	Project Initiation sign off	100.00%	2.00	0.00	Nikunj , Doshi	02/11/2021	02/11/2021	Complete
✎	Complete	2-Planning	1	Analysis	Project Vision Diagram	100.00%	4.00	0.00	Nikunj , Doshi	02/15/2021	02/18/2021	Complete
✎	Complete	2-Planning	2	Next Steps	Architecture Design	100.00%	4.00	0.00	Nikunj , Doshi	02/16/2021	02/19/2021	Complete
✎	Complete	2-Planning	2	Next Steps	Converting Functional Specs to Technical Specs	100.00%	4.00	0.00	Navaneeta, Naik	02/22/2021	02/24/2021	Complete

Velero Screenshot – Group Allocation 2



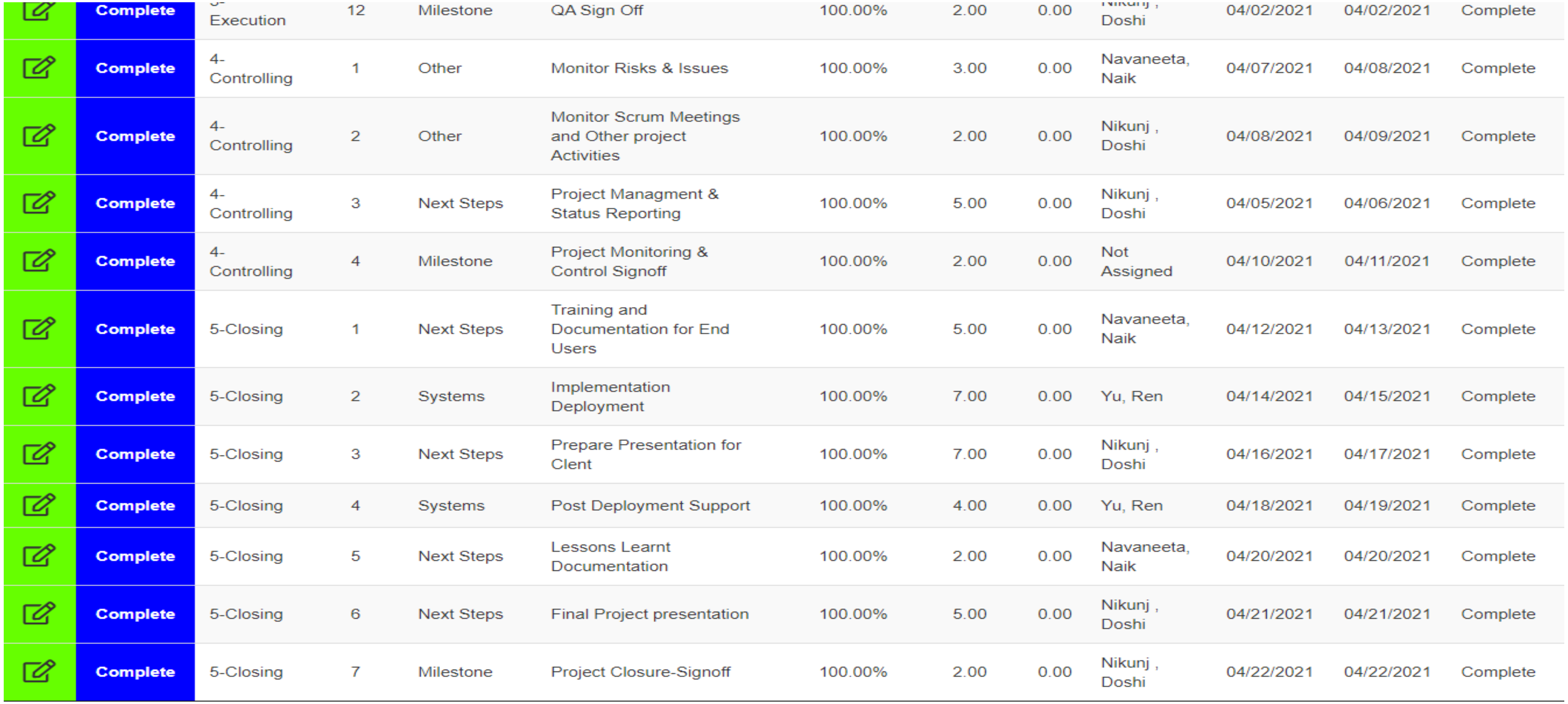
✎	**Complete**	2-Planning	2	Next Steps	Converting Functional Specs to Technical Specs	100.00%	4.00	0.00	Navaneeta, Naik	02/22/2021	02/24/2021	Complete
✎	**Complete**	2-Planning	3	Analysis	Knowing the right Databases	100.00%	4.00	0.00	Navaneeta, Naik	02/16/2021	02/19/2021	Complete
✎	**Complete**	2-Planning	4	Milestone	Architecture review & Approval	100.00%	2.00	0.00	Yu, Ren	02/22/2021	02/23/2021	Complete
✎	**Complete**	2-Planning	5	Next Steps	Identify Frameworks and Data Visualization	100.00%	2.00	0.00	Navaneeta, Naik	02/25/2021	02/26/2021	Complete
✎	**Complete**	2-Planning	7	Milestone	Project Planning Signoff	100.00%	2.00	0.00	Yu, Ren	02/27/2021	02/27/2021	Complete
✎	**Complete**	3-Execution	1	Analysis	Analyze the Data Set	100.00%	5.00	0.00	Navaneeta, Naik	03/01/2021	03/03/2021	Complete
✎	**Complete**	3-Execution	2	Development	Prepare Business Metadata	100.00%	4.00	0.00	Nikunj , Doshi	03/03/2021	03/05/2021	Complete
✎	**Complete**	3-Execution	2	Development	Configuring the Neo4j Environment Setup	100.00%	3.00	0.00	Not Assigned	03/04/2021	03/06/2021	Complete
✎	**Complete**	3-Execution	2	Development	Data Profiling	100.00%	8.00	0.00	Navaneeta, Naik	03/08/2021	03/12/2021	Complete
✎	**Complete**	3-Execution	2	Development	Data Validation and Data Visualization in Neo4j	100.00%	3.00	0.00	Yu, Ren	03/10/2021	03/12/2021	Complete
✎	**Complete**	3-Execution	3	Development	Load Sample Data in Neo4j	100.00%	4.00	0.00	Not Assigned	03/11/2021	03/12/2021	Complete
✎	**Complete**	3-Execution	4	Development	Main Dataset Load	100.00%	8.00	0.00	Yu, Ren	03/13/2021	03/14/2021	Complete
✎	**Complete**	3-Execution	4	Development	Data Cleaning & Wrangling in entire data	100.00%	4.00	0.00	Navaneeta, Naik	03/15/2021	03/16/2021	Complete

Velero Screenshot - Group Allocation 3



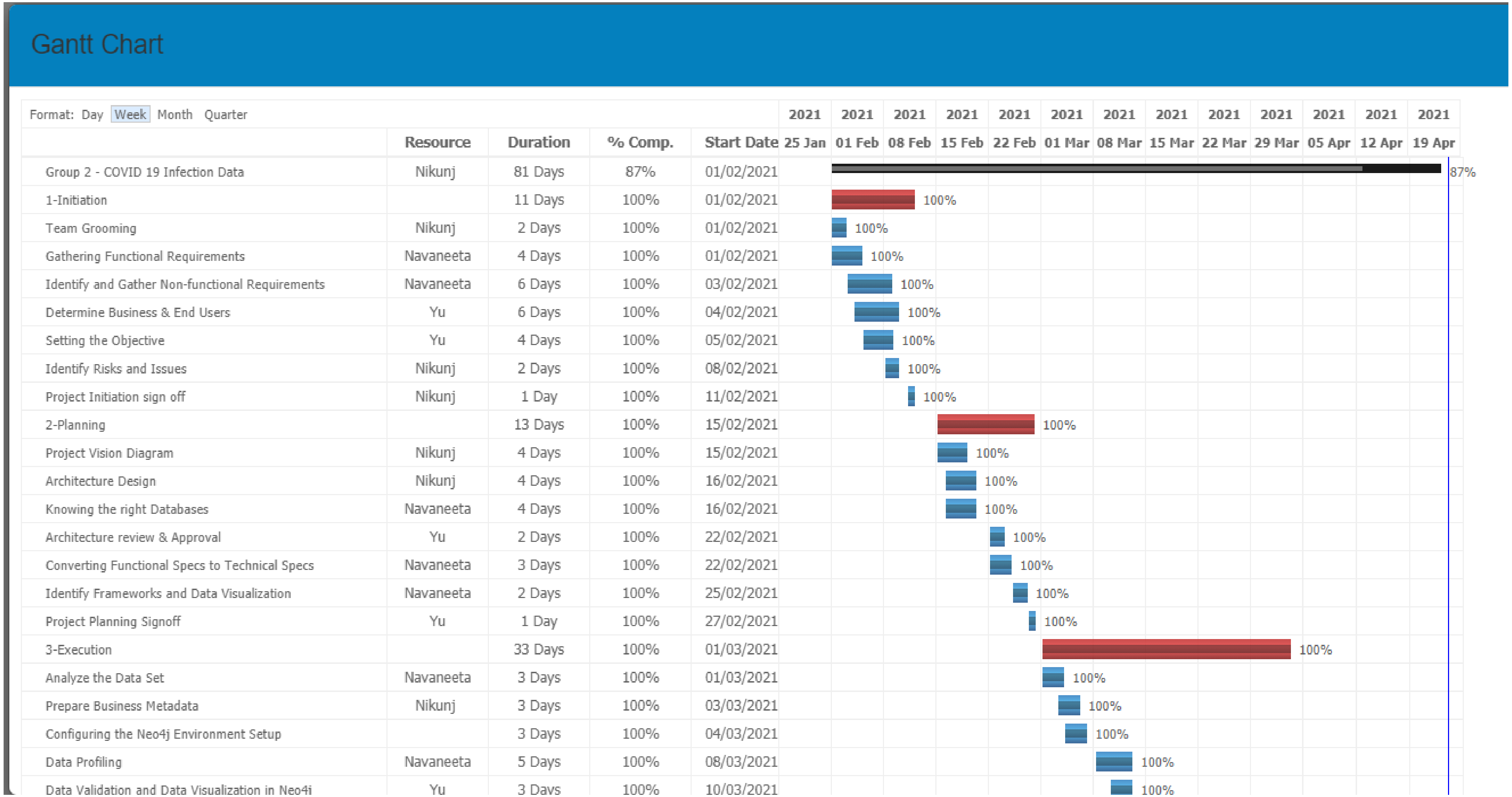
✎	**Complete**	3-Execution	4	Development	Writing the Business Metadata Terms	100.00%	3.00	0.00	Nikunj , Doshi	03/16/2021	03/18/2021	Complete
✎	**Complete**	3-Execution	5	Development	Unit Test	100.00%	2.00	0.00	Yu, Ren	03/19/2021	03/19/2021	Complete
✎	**Complete**	3-Execution	6	Development	QC for complete dataset	100.00%	1.00	0.00	Yu, Ren	03/20/2021	03/21/2021	Complete
✎	**Complete**	3-Execution	7	Development	Data Visualization Preparation and Development	100.00%	5.00	0.00	Navaneeta, Naik	03/21/2021	03/25/2021	Complete
✎	**Complete**	3-Execution	8	Development	Final Visualizations and Dashboard Generation	100.00%	3.00	0.00	Navaneeta, Naik	03/25/2021	03/27/2021	Complete
✎	**Complete**	3-Execution	9	QA	System Integration Testing	100.00%	3.00	0.00	Nikunj , Doshi	03/28/2021	03/29/2021	Complete
✎	**Complete**	3-Execution	10	QA	UAT Testing	100.00%	4.00	0.00	Nikunj , Doshi	03/30/2021	03/31/2021	Complete
✎	**Complete**	3-Execution	11	Milestone	Development Sign Off	100.00%	2.00	0.00	Yu, Ren	04/01/2021	04/01/2021	Complete
✎	**Complete**	3-Execution	12	Milestone	QA Sign Off	100.00%	2.00	0.00	Nikunj , Doshi	04/02/2021	04/02/2021	Complete
✎	**Complete**	4-Controlling	1	Other	Monitor Risks & Issues	100.00%	3.00	0.00	Navaneeta, Naik	04/07/2021	04/08/2021	Complete
✎	**Complete**	4-Controlling	2	Other	Monitor Scrum Meetings and Other project Activities	100.00%	2.00	0.00	Nikunj , Doshi	04/08/2021	04/09/2021	Complete
✎	**Complete**	4-Controlling	3	Next Steps	Project Managment & Status Reporting	100.00%	5.00	0.00	Nikunj , Doshi	04/05/2021	04/06/2021	Complete

Velero Screenshot – Group Allocation 4



✎	Complete	5- Execution	12	Milestone	QA Sign Off	100.00%	2.00	0.00	Nikunj , Doshi	04/02/2021	04/02/2021	Complete
✎	Complete	4- Controlling	1	Other	Monitor Risks & Issues	100.00%	3.00	0.00	Navaneeta, Naik	04/07/2021	04/08/2021	Complete
✎	Complete	4- Controlling	2	Other	Monitor Scrum Meetings and Other project Activities	100.00%	2.00	0.00	Nikunj , Doshi	04/08/2021	04/09/2021	Complete
✎	Complete	4- Controlling	3	Next Steps	Project Managment & Status Reporting	100.00%	5.00	0.00	Nikunj , Doshi	04/05/2021	04/06/2021	Complete
✎	Complete	4- Controlling	4	Milestone	Project Monitoring & Control Signoff	100.00%	2.00	0.00	Not Assigned	04/10/2021	04/11/2021	Complete
✎	Complete	5-Closing	1	Next Steps	Training and Documentation for End Users	100.00%	5.00	0.00	Navaneeta, Naik	04/12/2021	04/13/2021	Complete
✎	Complete	5-Closing	2	Systems	Implementation Deployment	100.00%	7.00	0.00	Yu, Ren	04/14/2021	04/15/2021	Complete
✎	Complete	5-Closing	3	Next Steps	Prepare Presentation for Clent	100.00%	7.00	0.00	Nikunj , Doshi	04/16/2021	04/17/2021	Complete
✎	Complete	5-Closing	4	Systems	Post Deployment Support	100.00%	4.00	0.00	Yu, Ren	04/18/2021	04/19/2021	Complete
✎	Complete	5-Closing	5	Next Steps	Lessons Learnt Documentation	100.00%	2.00	0.00	Navaneeta, Naik	04/20/2021	04/20/2021	Complete
✎	Complete	5-Closing	6	Next Steps	Final Project presentation	100.00%	5.00	0.00	Nikunj , Doshi	04/21/2021	04/21/2021	Complete
✎	Complete	5-Closing	7	Milestone	Project Closure-Signoff	100.00%	2.00	0.00	Nikunj , Doshi	04/22/2021	04/22/2021	Complete

Velero Screenshot - Gantt Chart 1

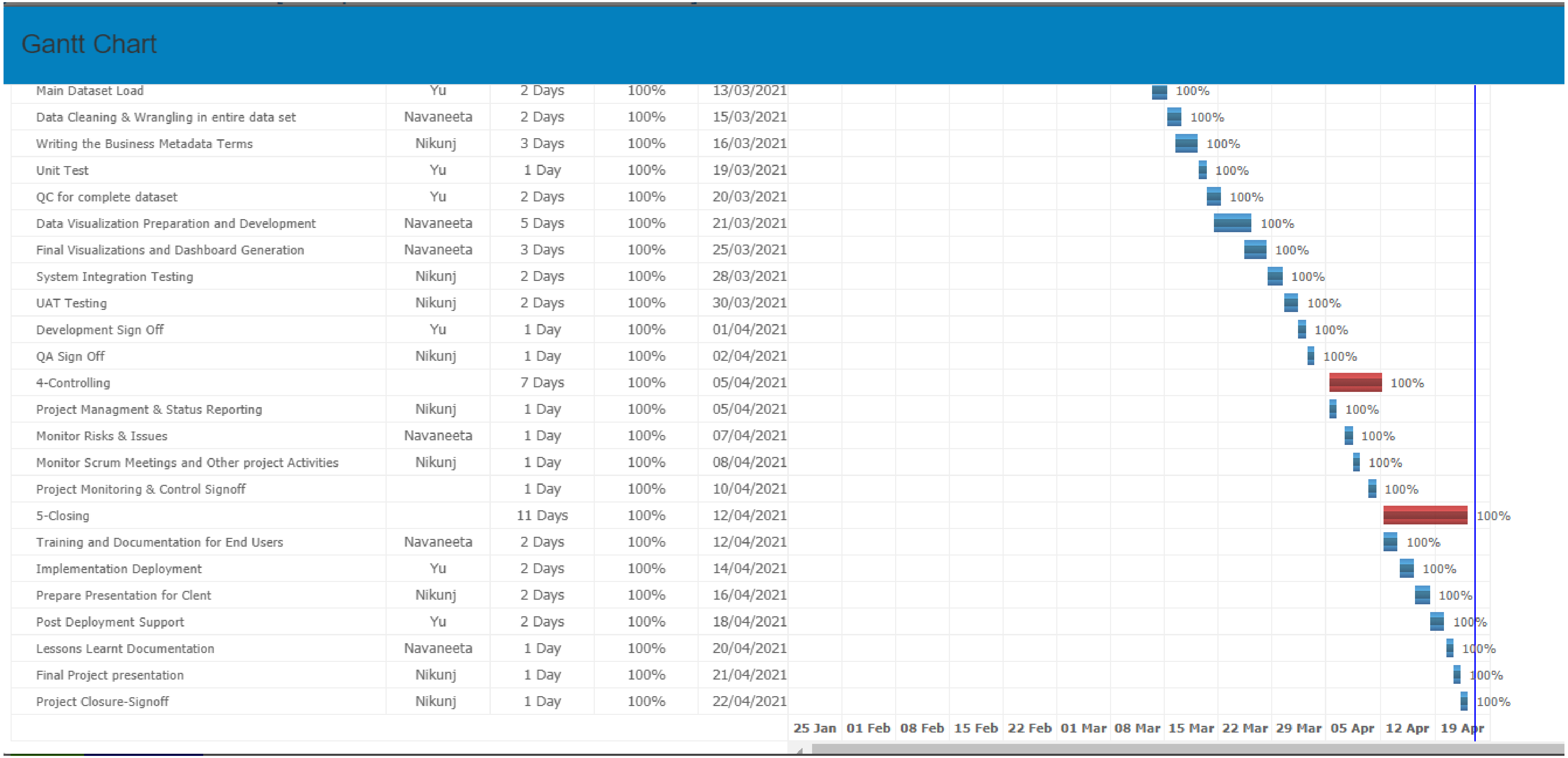


Gantt Chart

Format: Day Week Month Quarter

	Resource	Duration	% Comp.	Start Date
Group 2 - COVID 19 Infection Data	Nikunj	81 Days	87%	01/02/2021
1-Initiation		11 Days	100%	01/02/2021
Team Grooming	Nikunj	2 Days	100%	01/02/2021
Gathering Functional Requirements	Navaneeta	4 Days	100%	01/02/2021
Identify and Gather Non-functional Requirements	Navaneeta	6 Days	100%	03/02/2021
Determine Business & End Users	Yu	6 Days	100%	04/02/2021
Setting the Objective	Yu	4 Days	100%	05/02/2021
Identify Risks and Issues	Nikunj	2 Days	100%	08/02/2021
Project Initiation sign off	Nikunj	1 Day	100%	11/02/2021
2-Planning		13 Days	100%	15/02/2021
Project Vision Diagram	Nikunj	4 Days	100%	15/02/2021
Architecture Design	Nikunj	4 Days	100%	16/02/2021
Knowing the right Databases	Navaneeta	4 Days	100%	16/02/2021
Architecture review & Approval	Yu	2 Days	100%	22/02/2021
Converting Functional Specs to Technical Specs	Navaneeta	3 Days	100%	22/02/2021
Identify Frameworks and Data Visualization	Navaneeta	2 Days	100%	25/02/2021
Project Planning Signoff	Yu	1 Day	100%	27/02/2021
3-Execution		33 Days	100%	01/03/2021
Analyze the Data Set	Navaneeta	3 Days	100%	01/03/2021
Prepare Business Metadata	Nikunj	3 Days	100%	03/03/2021
Configuring the Neo4j Environment Setup		3 Days	100%	04/03/2021
Data Profiling	Navaneeta	5 Days	100%	08/03/2021
Data Validation and Data Visualization in Neo4i	Yu	3 Davs	100%	10/03/2021

Velero Screenshot - Gantt Chart 2

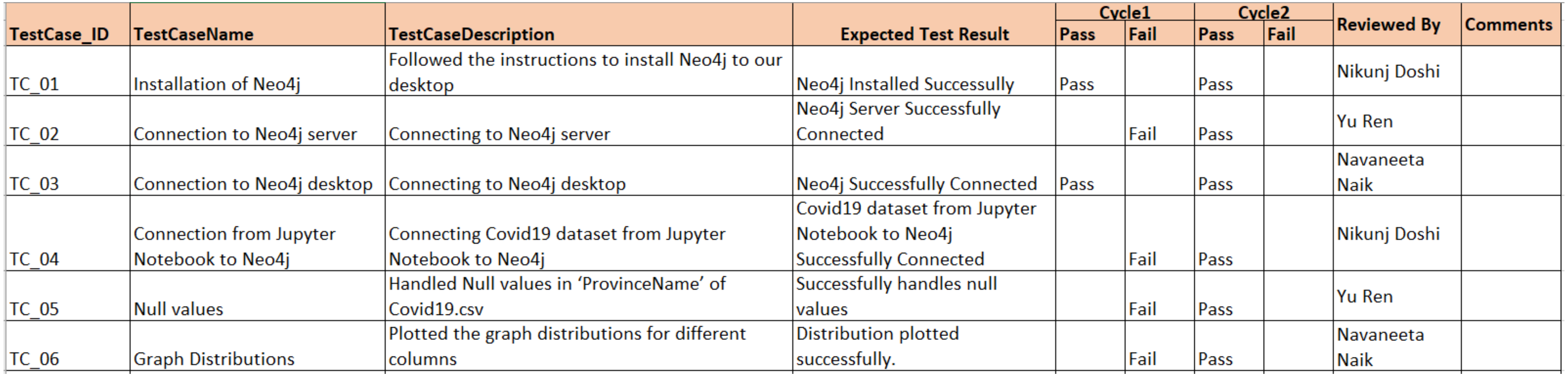


Gantt Chart

Main Dataset Load	Yu	2 Days	100%	13/03/2021
Data Cleaning & Wrangling in entire data set	Navaneeta	2 Days	100%	15/03/2021
Writing the Business Metadata Terms	Nikunj	3 Days	100%	16/03/2021
Unit Test	Yu	1 Day	100%	19/03/2021
QC for complete dataset	Yu	2 Days	100%	20/03/2021
Data Visualization Preparation and Development	Navaneeta	5 Days	100%	21/03/2021
Final Visualizations and Dashboard Generation	Navaneeta	3 Days	100%	25/03/2021
System Integration Testing	Nikunj	2 Days	100%	28/03/2021
UAT Testing	Nikunj	2 Days	100%	30/03/2021
Development Sign Off	Yu	1 Day	100%	01/04/2021
QA Sign Off	Nikunj	1 Day	100%	02/04/2021
4-Controlling		7 Days	100%	05/04/2021
Project Managment & Status Reporting	Nikunj	1 Day	100%	05/04/2021
Monitor Risks & Issues	Navaneeta	1 Day	100%	07/04/2021
Monitor Scrum Meetings and Other project Activities	Nikunj	1 Day	100%	08/04/2021
Project Monitoring & Control Signoff		1 Day	100%	10/04/2021
5-Closing		11 Days	100%	12/04/2021
Training and Documentation for End Users	Navaneeta	2 Days	100%	12/04/2021
Implementation Deployment	Yu	2 Days	100%	14/04/2021
Prepare Presentation for Clent	Nikunj	2 Days	100%	16/04/2021
Post Deployment Support	Yu	2 Days	100%	18/04/2021
Lessons Learnt Documentation	Navaneeta	1 Day	100%	20/04/2021
Final Project presentation	Nikunj	1 Day	100%	21/04/2021
Project Closure-Signoff	Nikunj	1 Day	100%	22/04/2021

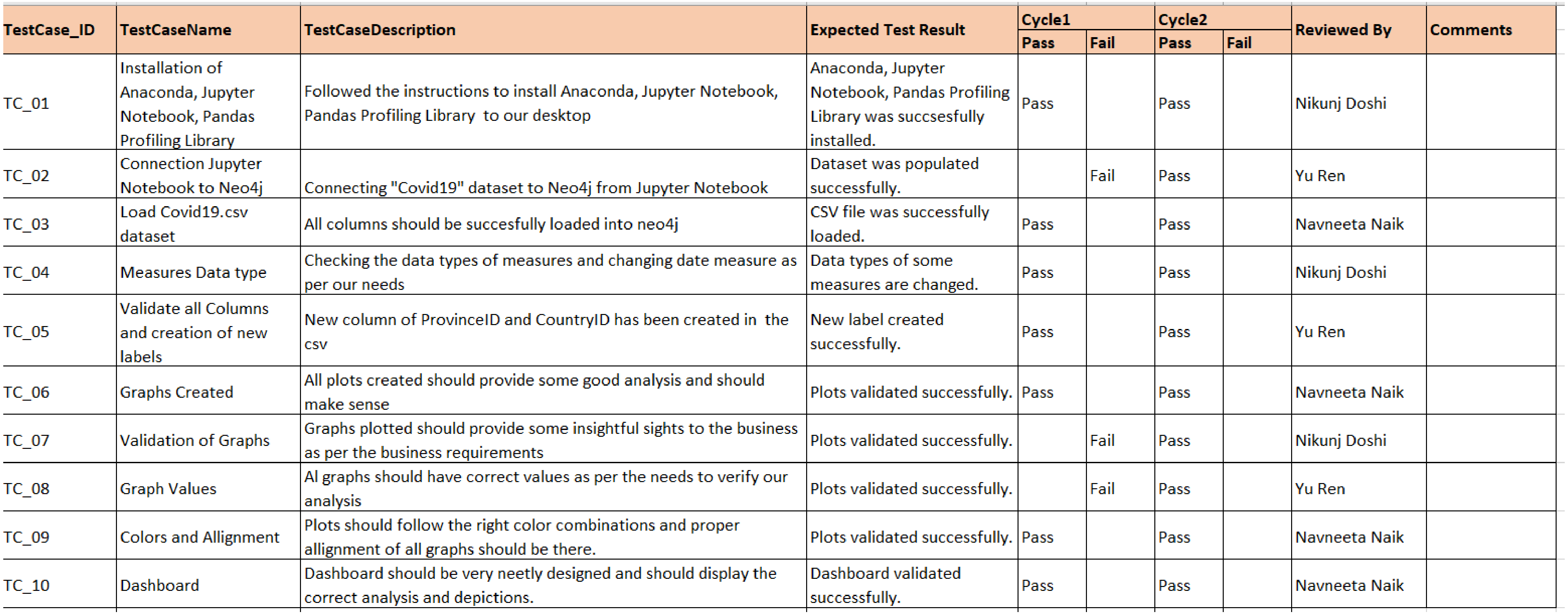
Q/A – TEST CASES

Unit Test Cases - Neo4j



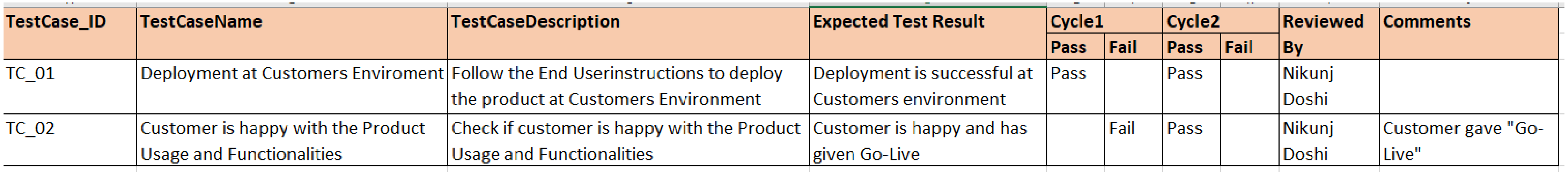
TestCase_ID	TestCaseName	TestCaseDescription	Expected Test Result	Cycle1		Cycle2		Reviewed By	Comments
				Pass	Fail	Pass	Fail		
TC_01	Installation of Neo4j	Followed the instructions to install Neo4j to our desktop	Neo4j Installed Successully	Pass		Pass		Nikunj Doshi	
TC_02	Connection to Neo4j server	Connecting to Neo4j server	Neo4j Server Successfully Connected		Fail	Pass		Yu Ren	
TC_03	Connection to Neo4j desktop	Connecting to Neo4j desktop	Neo4j Successfully Connected	Pass		Pass		Navaneeta Naik	
TC_04	Connection from Jupyter Notebook to Neo4j	Connecting Covid19 dataset from Jupyter Notebook to Neo4j	Covid19 dataset from Jupyter Notebook to Neo4j Successfully Connected		Fail	Pass		Nikunj Doshi	
TC_05	Null values	Handled Null values in 'ProvinceName' of Covid19.csv	Successfully handles null values		Fail	Pass		Yu Ren	
TC_06	Graph Distributions	Plotted the graph distributions for different columns	Distribution plotted successfully.		Fail	Pass		Navaneeta Naik	

System and Integration Testing



TestCase_ID	TestCaseName	TestCaseDescription	Expected Test Result	Cycle1		Cycle2		Reviewed By	Comments
				Pass	Fail	Pass	Fail		
TC_01	Installation of Anaconda, Jupyter Notebook, Pandas Profiling Library	Followed the instructions to install Anaconda, Jupyter Notebook, Pandas Profiling Library to our desktop	Anaconda, Jupyter Notebook, Pandas Profiling Library was succsesfully installed.	Pass		Pass		Nikunj Doshi	
TC_02	Connection Jupyter Notebook to Neo4j	Connecting "Covid19" dataset to Neo4j from Jupyter Notebook	Dataset was populated successfully.		Fail	Pass		Yu Ren	
TC_03	Load Covid19.csv dataset	All columns should be succesfully loaded into neo4j	CSV file was successfully loaded.	Pass		Pass		Navneeta Naik	
TC_04	Measures Data type	Checking the data types of measures and changing date measure as per our needs	Data types of some measures are changed.	Pass		Pass		Nikunj Doshi	
TC_05	Validate all Columns and creation of new labels	New column of ProvinceID and CountryID has been created in the csv	New label created successfully.	Pass		Pass		Yu Ren	
TC_06	Graphs Created	All plots created should provide some good analysis and should make sense	Plots validated successfully.	Pass		Pass		Navneeta Naik	
TC_07	Validation of Graphs	Graphs plotted should provide some insightful sights to the business as per the business requirements	Plots validated successfully.		Fail	Pass		Nikunj Doshi	
TC_08	Graph Values	Al graphs should have correct values as per the needs to verify our analysis	Plots validated successfully.		Fail	Pass		Yu Ren	
TC_09	Colors and Allignment	Plots should follow the right color combinations and proper allignment of all graphs should be there.	Plots validated successfully.	Pass		Pass		Navneeta Naik	
TC_10	Dashboard	Dashboard should be very neetly designed and should display the correct analysis and depictions.	Dashboard validated successfully.	Pass		Pass		Navneeta Naik	

User Acceptance Testing



TestCase_ID	TestCaseName	TestCaseDescription	Expected Test Result	Cycle1		Cycle2		Reviewed By	Comments
				Pass	Fail	Pass	Fail		
TC_01	Deployment at Customers Enviroment	Follow the End Userinstructions to deploy the product at Customers Environment	Deployment is successful at Customers environment	Pass		Pass		Nikunj Doshi	
TC_02	Customer is happy with the Product Usage and Functionalities	Check if customer is happy with the Product Usage and Functionalities	Customer is happy and has given Go-Live		Fail	Pass		Nikunj Doshi	Customer gave "Go-Live"

THANK YOU