report

by Sristi Sristi

Submission date: 14-May-2020 03:16PM (UTC+0530)
Submission ID: 1324023489
File name: sheishti-reeport.pdf (1,007.52K)
Word count: 10393
Character count: 52797

PROJECT REPORT

ON

Industrial Internship

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted By

Name: Shristi Saxena

ROLL NO.: 160105015

Batch: 2016-20

Internal Mentor

Name: Dr. Rama Sushil

Post: Professor

Department: IT

Department of Information Technology

DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2019-20

DECLARATION

I hereby certify that the work, which is being presented in the report/ project report, entitled **"Industrial Internship"** with three projects i.e. 'Search & NLP', 'Glue Migration', 'Covid-19 Application', in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of my/our own work carried out during the period *from* 6^{th} *January, 2020* to *to* 3^{rd} *July, 2020* under the supervision of **Dr. Rama Sushil.**

Date: **Signature of the Candidates**

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: **Signature of the Supervisor**

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: **Signature of the COE**

CERTIFICATE

This is to certify that the Project entitled **"Industrial Internship"** with three projects i.e. 'Search & NLP', 'Glue Migration', 'Covid-19 Application', in partial fulfillment of the requirement for the award of the **Degree B.Tech** in **Information Technology,** submitted to **DIT University, Dehradun, Uttarakhand, India** is an authentic record of bona fide research work carried out by **Shristi Saxena,** Roll No - 160105015 under my supervision and guidance.

Signature of Guide

Name
Professor
Department of
DIT University

Signature of HOD

Dr. Anil km. Dahiya
Head of Department
Department of IT
DIT University

ACKNOWLEDGEMENT

We are over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these efforts and knowledge, well above the level of simplicity and into something concreate. I would like to express my sincere thanks of gratitude to my mentor Dr. Rama Sushil for her immense support and guidance as well as our Project Coordinator Nitin Thapliyal and HOD IT Dr. Anil Kumar Dhaiya who gave us the golden opportunity to do this wonderful project on the topic title, which also helped us in doing a lot of research and I came to know about so many new things, I would also like to thank ZS Associates which gave such interesting projects which enabled me to learn new technologies.

Shristi Saxena

ABSTRACT

In Industrial Internship, I got opportunity to work over 3 different projects which made me learn various Technologies. First project is entitled as “Search & NLP”, it is an enterprise platform used to discover information and extract previously unattainable insights from structured and unstructured data. By leveraging common platform components including advanced search and discovery capabilities coupled with NLP techniques, Client users can uncover valuable information expediently.

Second project is entitled as “Glue Migration”, in this we were migrating our clients all historical data from the previous data source that was HIVE to GLUE and crawling all the new data coming to Glue. We are not only moving data to the source i.e. Glue but also validating the schema of data table post migration.

Third project is entitled as “Covid-19 Application”, wherein we are creating an application for our Pharmaceutical client, which will help them to effectively plan their medicinal production and supply of same. The application has all data of Covid-19 affected areas from all over the world and gives a projection to the clients with reports which suggests them how their Production and Supply Chain can be affected from Covid affected areas.

My role in all the three projects were to Test the whole performance and working of the projects including the User Interface and backend. The process followed by me to perform Testing was I need to maintain a Testing Scenario Sheet along with a Test Cases Sheet which contains the flow of Testing and status of each Cases whether Passed or Failed.

TABLE OF CONTENT

S. No.	Topics	Page No.
1.	Search & NLP	1
1.1	Overview	1
1.2	Purpose	2
1.3	Motivation	2
1.4	Objective	3
1.5	Features	4
1.6	ER Diagram	5
1.7	Implementation Details	6
1.7.1	Software Implementation	6
1.8	Methodology	11
1.9	Screenshots	14
1.9.1	Solr Displaying Data	14
1.9.2	Amazon S3 View	15
2.	Glue Migration	16
2.1	Overview	16
2.2	Purpose	17
2.3	Motivation	18
2.4	Objective	18
2.5	ER Diagram	19
2.6	Implementation Details	20
2.6.1	Software Implementation	20
2.7	Methodology	23
2.8	Screenshots	25
2.8.1	Airflow UI with all DAGs & their information	25
2.8.2	Airflow DAG	26

3.	COVID-19 Application	27
3.1	Overview	27
3.2	Purpose	28
3.3	Motivation	28
3.4	Objective	29
3.5	ER Diagram	30
3.6	Implementation Details	32
3.6.1	Software Implementation	32
3.7	Methodology	37
3.8	Screenshots	40
3.8.1	Solr Displaying Data View	40
3.8.2	Amazon S3 View	41
	Reference	43

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
1.1	ER Diagram of Search & NLP	5
1.2	Solr Displaying Data	14
1.3	Amazon S3 View	15
2.1	ER Diagram of Glue Migration	19
2.2	Airflow UI with all DAGs & their information	25
2.3	Airflow DAG	26
3.1	ER Diagram of COVID-19 Application	31
3.2	Solr Displaying Data	40
3.3	Amazon S3 View	41
3.4	Kepler Demo View	42

CHAPTER 1: SEARCH & NLP

Search & NLP project is a searching web application. It is an appropriate and useful web application for managing and viewing a large amount of current and historic data for various medicines manufactured by the Client and diseases which can be cured by those medicines.

1.1 OVERVIEW

This project is a searching web application. It is an appropriate and useful web application for managing and viewing a large amount of current and historic data for various medicines manufactured by the Client and diseases which can be cured by those medicines. It is a common platform for different users of Client, for searching and analysis over the data, to view statics of data on various time periods. Search and Natural Language Processing (NLP) is an enterprise platform used to discover information and extract previously unattainable insights from structured and unstructured data.

By leveraging common platform components including advanced search and discovery capabilities coupled with NLP techniques, Client users can uncover valuable information expediently. Custom built crawlers and connectors extract centrally store and index data for search use cases. Crawlers are primarily used to fetch current data but also used to fetch Data from old sources where data was previously kept i.e. HBase, HDFS (Attachments), Old Solr. Client users can uncover valuable information expediently. Custom built crawlers and connectors extract centrally store and index data for search use cases.

Login module has a section for registered users to login into the web-based UI application which shows complete information of diseases and their related medicines. The Home Page Module is the module to which user interacts the first as soon as he logins to the UI. Dashboard module has the bar or line graphical representation of the data present in the result asset and has a Pie-chart representation of the same.

There is an option to put Time-Interval as a filter whose different values are (Past-Week, Past-Month, Past-Quarter, Past-Year). Survey Page has all the elements same as Home Page the only and major difference between them is Home Page has all type of results whereas Survey page has only the Survey data that has previously collected for various corresponding diseases.

Export page exhibits features of exporting the result asset in the form of an Excel sheet. Settings provides all personalized settings to the user, like to choose the Filters they want to have on Filter Facet, to set the type of View from the list of views (List, Grid, Table) and to choose the set of columns user wants in the Table view.

1.2 PURPOSE

Search and Natural Language Processing (NLP) is an enterprise platform used to discover information and extract previously unattainable insights from structured and unstructured data. It is an appropriate and useful web application for managing and viewing a large amount of current and historic data for various medicines manufactured by the Client and diseases which can be cured by those medicines. It is a common platform for different users of Client, for searching and analysis over the data, to view statics of data on various time periods. By leveraging common platform components including advanced search and discovery capabilities coupled with NLP techniques, Client users can uncover valuable information expediently.

1.3 MOTIVATION

Search Application was required by Clients to have data regarding all the medicines they manufacture and supply in-according to their therapeutic areas they deal with. The application has data about all medicines manufactured by them. The data preparation platform provides durable components that prepare data for advanced analytics usage. The Natural Language Processing (NLP) platform provides durable components that enable advanced language features like unsupervised and semi-supervised topic detection.

Sentiment analysis and entity extraction to help derive meaningful insights. The data preparation platform provides durable components that prepare data for advanced analytics usage. These services include document conversion, OCR, reference data integration and de-identification. Building a self-service portal to help users surface insights from a corpus of data that can be uploaded via a simple user interface.

1.4 OBJECTIVE

Search application is an appropriate and useful web application for managing and viewing a large amount of current and historic data for various medicines manufactured by the Client and diseases which can be cured by those medicines. It is a common platform for different users of Client, for searching and analysis over the data, to view statics of data on various time periods. By leveraging common platform components including advanced search and discovery capabilities coupled with NLP techniques, Client users can uncover valuable information expediently. Search and Natural Language Processing (NLP) is an enterprise platform used to discover information and extract previously unattainable insights from structured and unstructured data. It is a common platform for different users of Client, for searching and analysis over the data, to view statics of data on various time periods.

Login module has a section for registered users to login into the web-based UI application which shows complete information of diseases and their related medicines. The Home Page Module is the module to which user interacts the first as soon as he logins to the UI. Dashboard module has the bar or line graphical representation of the data present in the result asset and has a Pie-chart representation of the same, with which there is an option to put Time-Interval as a filter whose different values are (Past-Week, Past-Month, Past-Quarter, Past-Year). Survey Page has all the elements same as Home Page the only and major difference between them is Home Page has all type of results whereas Survey page has only the Survey data that has previously collected for various corresponding diseases.

Export page exhibits features of exporting the result asset in the form of an Excel sheet. Settings provides all personalized settings to the user, like to choose the Filters they want to have on Filter Facet, to set the type of View from the list of views (List, Grid, Table) and to choose the set of columns user wants in the Table view.

1.5 FEATURES of SEARCH & NLP

a. Multi-Channel Ingestion Service

The multi-channel ingestion service receives data from multiple sources including Client repositories, social media and external websites. Custom built crawlers and connectors extract centrally store and index data for search use cases.

b. Data Preparation for Advanced Analytics

The data preparation platform provides durable components that prepare data for advanced analytics usage. These services include document conversion, OCR, reference data integration and de-identification.

c. Enables NLP

The Natural Language Processing (NLP) platform provides durable components that enable advanced language features like unsupervised and semi-supervised topic detection, sentiment analysis and entity extraction to help derive meaningful insights.

d. Data Consumption

Data consumption is enabled through the search portal, virtual assistants and chatbots. To make sure that the Application's performance is optimized the backend uses GraphQL rather than Rest APIs.

1.6 ER Diagram

Fig. 1.1 represents the ER Diagram of Search & NLP, in which the user will put his credentials i.e. Id and Password into the Login Page for user authentication. On successful login user will be led to the Home Page which has three parts Result Asset, Dashboard Page and Survey Page. The Home Page has various filters that can govern the search and results of Search and Filters. The Dashboard has Graphical representation of all the data and Reports around that graphical data. The Survey page has all the surveys collected from various Patients who are using Clients Products. There is an Export Feature provided from Home Page Result Asset and Survey Page Result Asset.

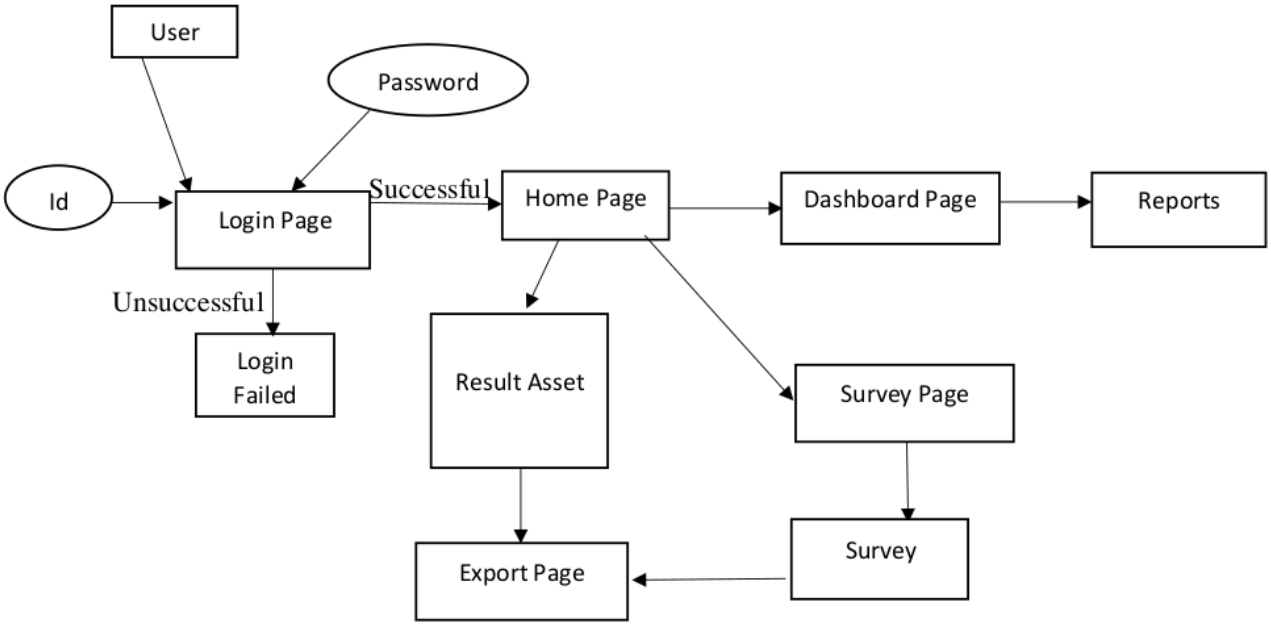

Fig 1.1 ER Diagram of Search & NLP

1.7 IMPLEMENTATION DETAILS

This section contains a brief note of all the technologies used in the development of this project for Frontend and Backend. The Hardware requirements used in developing the Search Application is Laptop or Desktop with RAM of 4GB and above. Below all the Software requirements are mentioned required for developing the Search Application.

1.7.1 Software Interface

Languages	:	Python, GraphQL
Front End	:	HTML, CSS, Bootstrap, JavaScript, NodeJS
Platform	:	Pycharm, Web browsers (Chrome, Firefox, Internet Explorer)
Backend	:	Python
Databases	:	DynamoDB, Solr, Amazon S3

- **GraphQL**

GraphQL is a query language for APIs, and a runtime for your current data to satisfy those queries. GraphQL offers a detailed and comprehensible summary of the data in your API, allows consumers the opportunity to ask for exactly what they need and nothing more, makes APIs simpler to grow over time and unlocks powerful development tools.

It is an open- sourced search engine that stores data in tokens type in containers. It offers an approach to the creation of web APIs and was compared with REST and other architectures for web services. It allows clients to specify the structure from the server, therefore preventing the return of overly large quantities of data, but this has consequences for how efficient web caching of query results can be. The versatility and richness of the query language also adds complexity for simple APIs.

- **Python**

It is the high level of dynamic semantics, interpreted, and oriented programming language. Its high degree of compatibility is built into data structures in conjunction with dynamic typing and scripting. The plain, easy-to-learn syntax of Python underlines readability and thus reduces the maintenance costs. The module and package that fosters software modularity and code reusability are provided by Python. The Python interpreter and its broad standard library can be distributed free of charge in source or binary form for all major platforms.

- **Pycharm**

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features. We still come across code we 're not sure about. Code to other people. Heck, our technology too. Often, for a purpose, we just want the arguments. Many times, we would like to learn args from positional or keyword. Or the Argument Types. Or its ideals by default. Or a good docstring, made. First, Quick Documentation (Ctrl-P Win / Linux, F1 macOS) brings non-obtrusive inline popup displaying all that information with a hyperlink where you can navigate to the definition. First, Fast Documentation (Ctrl-P Win / Linux, F1 macOS) brings non-obtrusive inline popup displaying all of that detail, with a hyperlink where you can navigate to the description. Press it again, and the popup becomes a tool window that updates for every symbol you land on. Still there, still supporting ... as with every other IDE tool slot, before you want to cover it.

PyCharm offers some of the best features to its users and developers in the following aspects –

- Code completion and inspection and Advanced debugging
- Support for web programming and frameworks such as Django and Flask

- **HTML**

The basic language for mark-ups for the documents intended to appear on a web-browser is the hypertext mark-up language (HTML). Technologies like the cascade of style sheets (CSS) and scripting languages like JavaScript can be used to help this. HTML documents are downloaded from a web server or local storage from Web browsers and translated to the web pages of multimedia. HTML explains the web page layout in a comprehensive manner and the documents originally included. Building blocks of HTML pages are HTML elements. The made page can be incorporated with HTML builds, pictures and other objects such as interactive forms. For text, including headings, sections, lists, links, quotations, and other things HTML offers a way to produce organized documents by defining structural semanticity. HTML elements are labelled, written with angle brackets. Tags like and include material directly in the tab. Sub-elements may also include other tags, such as surround and provide information on document texts. The HTML tags are not viewed by the browsers but are used by them to view the page content.

- **CSS**

Cascading Style Sheets (CSS) is a style plate term used to characterize the application of a markup document such as HTML. This separation will increase usability of contents, provide greater flexibility and control over the presentation characteristics, allow multi-web sites to share the formatted content by deciding which one is the basis. The CSS is a key technology in the World Wide Web alongside HTML and JavaScript. CSS for separating presentation and contents including templates, colors and font. Separation of formatting and contents also allows to display a single markup page in different styles for different rendering methods , e.g. on-screen, in print, by voice, or on Braille- based tactile tools. If content is viewed on a mobile device, CSS also has alternative formatting guidelines.

The name cascading derives from the priority scheme to decide the rule of style if more than one rule fits a particular feature. This goal is predictable in cascading.

- **Node.js**

It is cross-platformed for runtime environment of JavaScript that executes codes JavaScript outside web browser and open-sourced. It allows developers to use JavaScript to write command line programs, and server-side scripting codes that run on server-side scripts to generate dynamic web page content before the web browser is sent to the user. Node.js reflects a "JavaScript everywhere" unifying the development of web-applications around a common programming language, rather than separate languages for server-client-side scripts. It allows developers to use JavaScript to write command line programs, and server-side scripting codes that run on server-side scripts to generate dynamic web page content before the web browser is sent to the user.

- **DynamoDB**

Amazon DynamoDB is a fully managed NoSQL data base service that supports data structures and key values and is supported by Amazon.com under the Amazon Web Services portfolio. DynamoDB exposes and derives its name from Dynamo to a similar data model which has another underlying implementation. Dynamo has a multi-master architecture which requires clients to resolve version conflicts and for high durability and availability DynamoDB uses synchronous replication across multiple data centres. On 18 January 2012, DynamoDB was announced as an evolution of Amazon SimpleDB solution by Amazon CTO Werner Vogels. It was released. Dynamo has a multi-master architecture which requires clients to resolve version conflicts and for high durability and availability DynamoDB uses synchronous replication across multiple data centres. DynamoDB exposes and derives its name from Dynamo to a similar data model which has another underlying implementation.

- **Solr**

Solr (pronounced "solar") is an Apache Lucene project's open-source corporate search engine written in Java. Highlights, facet-level search, real-time indexing, dynamic clustering, data base integration, NoSqL functionality, as well as comprehensive handling of documents (for example Word or PDF) are all functionality of the business. Solr is optimized for scalability and fault tolerance to provide distributed search and index replication. Solr has an active developer community with regular releases and is widely used for corporate search and analytics. The Solr search server is operating autonomously. It utilizes the full-text indexing and search library for Lucene Java, and it has HTTP/XML and JSON-like REST APIs that support it in the most common programming languages. The external Solr software enables it to be customized without Java coding to several applications and has a plugin architecture that facilitates more customization. The same Apache Software Foundation engineering team produces Apache Lucene and Apache Solr. The Solr search server is operating autonomously. It utilizes the full-text indexing and search library for Lucene Java, and it has HTTP/XML and JSON-like REST APIs that support it in the most common programming languages.

- **Amazon S3**

Amazon Simple Storage Service provides storage of objects through cloud services. It is also known as Amazon S3 which is an Amazon Web Service. Amazon S3 uses the global e-commerce network with the same flexible storage system that Amazon.com does. The Amazon S3 can be used for storing items of any kind, such as internet storage, security and recovery, disaster recovery, collections of data, analytics data lakes, and hybrid cloud storage. AWS launched Amazon S3 on March 14, 2006 in the United States and in November 2007 in Europe. Amazon's easy storage service (Amazon S3) provides leading scalability, availability of content, protection, and efficiency in an artefact storage facility.

This helps consumers of all sizes and industries to store and secure any amount of data in a variety of uses, including websites, mobiles, backups and reconstruction, libraries, business apps, IoT devices and Big Data Analysis. Amazon S3 offers easy-to-use management functions, allowing you to arrange your data to comply with your particular sector, organization and conformity criteria and configure completed access control systems.

1.8 METHODOLOGY

This section contains the flow of the Search Application and all the modules included in this application are described in detail. This complete project is divided into different Modules as follows:

i. Login Module to Search UI.
ii. Home Page Module of Search UI.
iii. Dashboard Page Module of Search UI.
iv. Survey Page Module of Search UI.
v. Export Module of Search UI.
vi. Settings Module of Search UI.

i) Login Module:

This module has a section for registered users to login into the web-based UI application which shows complete information of diseases and their related medicines. This module also checks the authenticity of the user and also helps in case the user has forgotten his/her password to regain his password to account. In which the user will put his credentials i.e. Id and Password into the Login Page for user authentication. On successful login user will be led to the Home Page.

ii.) Home Page Module:

The Home Page Module is the module to which user interacts the first as soon as he logins to the UI, which contains the following assets:

1) Header: It contains all header elements of the UI.
   a) Search UI icon
   b) Searching Box Asset
   c) Save Search (To save any search made)
   d) View Search (To view any previously saved search)
   e) Bookmark icon (which opens a list of Bookmarked Assets)
   f) User Profile icon (which has Settings of UI & Log Out option)

2) Dashboard Page Button: A button that will lead to the Dashboard Page of UI.

3) Survey Page Button: A button that will lead to the Survey Page of UI.

4) Filter Asset: It has various filters that user can apply on the search made by him/her, then user will get results of their search with those filters applied on it.

5) Result Asset: This area contains all the results the UI has for the search made. It has three different type of Views i.e. Grid view, Table view, List view and has Pagination which changes according to count of results. The Home Page has various filters that can govern the search and results of Search and Filters.

iii.) Dashboard Page Module:

It has the bar or line graphical representation of the data present in the result asset and also has a Pie-chart representation of the same, with which there is an option to put Time-Interval as a filter whose different values are (Past-Week, Past-Month, Past-Quarter, Past-Year).

iv.) Survey Page Module:

It has all the elements same as Home Page the only and major difference between them is Home Page has all type of results whereas Survey page has only the Survey data that has previously collected for various corresponding diseases.

v.) Export Module:

There is an Export Feature provided from Home Page Result Asset and Survey Page Result Asset.
This page exhibits features of exporting the result asset in the form of an Excel sheet.

vi.) Setting Module:

It provides all personalized settings to the user, like to choose the Filters they want to have on Filter Facet, to set the type of View from the list of views (List, Grid, Table) and to choose the set of columns user wants in the Table view.

1.9 SCREENSHOTS

1.9.1 Solr Displaying Data as per query:

In the Fig 1.2 A Query is executed on Solr with a fq (filter query) is applied. Solr is used for data storage where user can not only store the data but has various and easy approaches to Query the data. The Data stored in Solr is stored into Collections and Aliases.

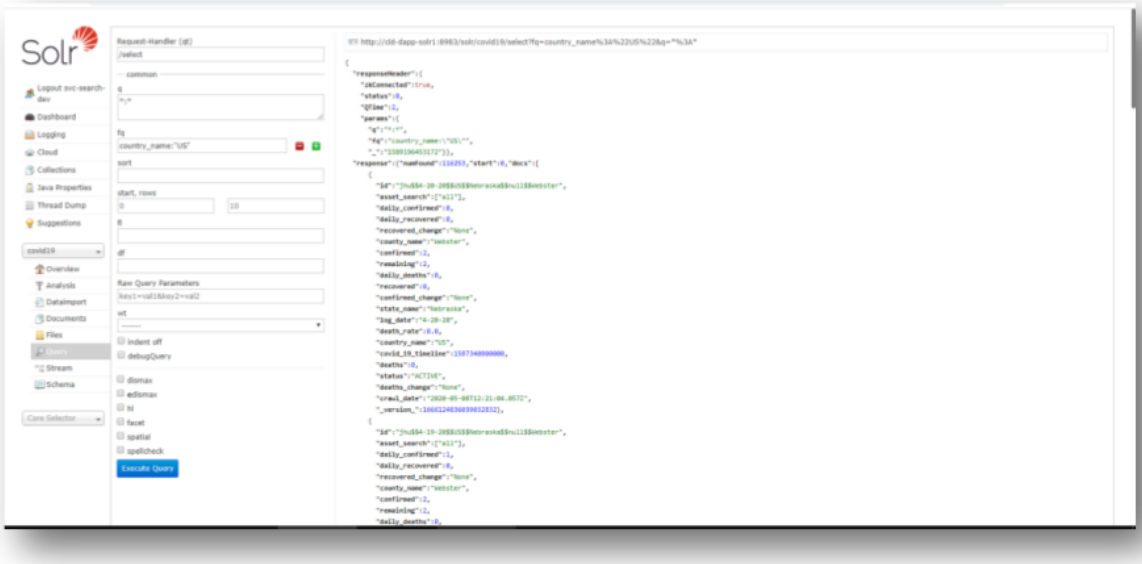

Fig 1.2 Solr Displaying Data as per query

1.9.2 Amazon S3 View

In the Fig. 1.3 the User Interface of Amazon S3 is shown, which stores data in form of Buckets. It not only provides feature of data storage in buckets but also allows user to perform general Operations on it.

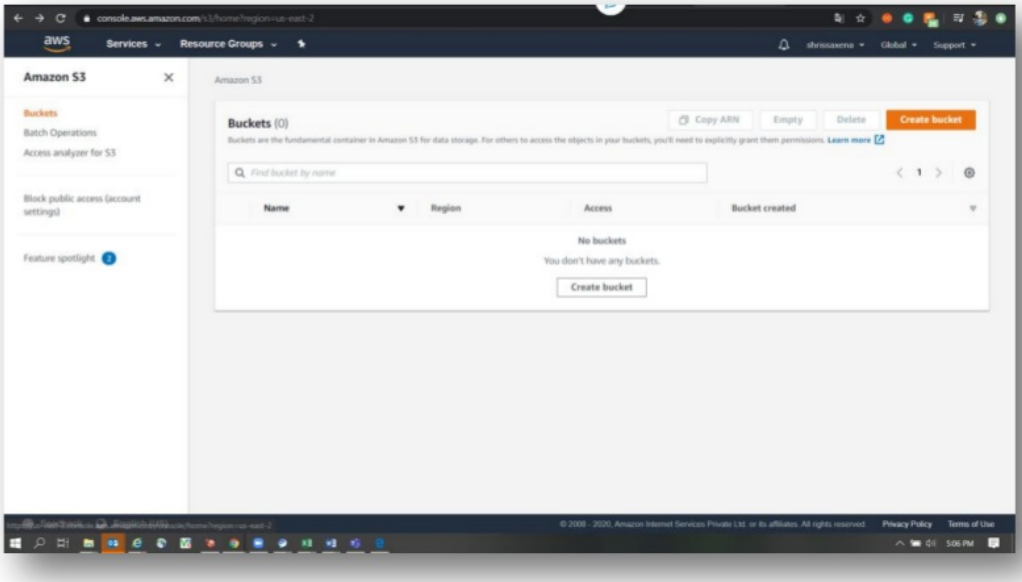

Fig 1.3 Amazon S3 view

CHAPTER 2: GLUE MIGRATION

Glue Migration is essentially a mechanism by which all existing data from the previous data source that were HIVE have migrated to the current GLUE data source and all new data have been moved to Glue. We do not just transfer data to the source, that is collect but also check the schemas of the data tables post migration.

2.1 OVERVIEW

Glue Migration is basically a process, in this we were migrating our clients all historical data from the previous data source that was HIVE to new data source GLUE and crawling all the new data coming to Glue. We are not only moving data to the source i.e. Glue but also validating the schema of data table post migration. It was initiated by our Client to modernize the data storage scheme as the technologies are evolving so they had a thought to change their previous data store Hive which gives an interface similar to SQL to query data stored in different databases across its file systems that are integrate with Hadoop, and Hadoop was not able to handle the frequent intake of data and was not cost efficient as well.

Hence it was required to switch to some other data store i.e. Glue which is a fully managed ETL service which allows customers to easily prepare and load their analysis data. The whole process of Migrating data from HIVE to GLUE is long and tough, hence these tasks are automated by using Airflow DAGS, wherein you can streamline all task by creating the DAGS by writing a Python Script which has configurations of that DAG and all its tasks are described with file paths in DAG configs. It is required to create schema for destination databases similar to source database's schema. Hence it was required to switch to some other data store i.e. Glue which is a fully managed ETL service which allows customers to easily prepare and load their analysis data.

It was initiated by our Client to modernize the data storage scheme as the technologies are evolving so they had a thought to change their previous data store Hive which gives an interface similar to SQL to query data stored in different databases across its file systems that are integrate with Hadoop.

Post migration all the data would be aligned accordingly and there would be no mis-migration of data. Migration process is performed by the Airflow DAGS, so we need to initiate its run from PuTTY the Unix terminal or from Airflow's UI. The Airflow UI shows all the states of the DAG run and notifies the user when the DAG run successfully completes and also when it fails. We need to have a comparison between the Source database's schema and destination database's schema, post migration process completes because it is possible that while migration some data might have missed and got failed to migrate hence it can be migrated again. We need to have a comparison between the Source data count and destination data count, post migration process completes because it is possible that while migration some data might have missed and got failed to migrate hence it can be migrated again.

2.2 PURPOSE

Glue Migration project was initiated by our Client to modernize the data storage scheme as the technologies are evolving so they had a thought to change their previous data store Hive which gives an interface similar to SQL to query data stored in different databases across its file systems that are integrate with Hadoop, and Hadoop was not able to handle the frequent intake of data and was not cost efficient as well.

Hence it was required to switch to some other data store i.e. Glue which is a fully managed ETL service which allows customers to easily prepare and load their analysis data. The AWS Management Console helps you to build and execute an ETL job with a few clicks. Just point AWS Glue to your AWS-stored data, and AWS Glue will discover and store the corresponding metadata.

Just point AWS Glue to your AWS-stored data, and AWS Glue will discover and store the corresponding metadata. The AWS Management Console helps you to build and execute an ETL job with a few clicks. Just point AWS Glue to your AWS-stored data, and AWS Glue will discover and store the corresponding metadata.

2.3 MOTIVATION

Our client initiated the Glue Migration project to upgrade its data storage scheme with emerging technology, which means that it thought of changing its previous datastore Hive which gives a SQL-like interface for querying data in various databases on Hadoop systems. Hadoop could not handle the popular data intake nor was it a co-operation. Therefore, some other data store had to be updated, i.e. Glue, a professionally operated ETL service that allows customers to plan and load their analysis data easily. You can create and perform an ETL job with a few clicks on the AWS management console, only point AWS to the data stored in your AWS and AWS will identify and store the metadata.

2.4 OBJECTIVE

Glue Migration is essentially a mechanism by which all existing data from the previous data source that were HIVE have moved to the current GLUE data source and all new data have been converted to Glue. We do not only transfer data to the source, that is. Collect but also check the post migration schedule of the data table. Our Client was introduced to update the data storage system as technology advance and had an idea of changing their previous Hive data store which provides the SQL-like interface for querying data stored in various databases through their Hadoop network and Hadoop was unable and not cost-effective to handle a regular intake of information. It was then appropriate to move to a separate data centre, i.e. Glue is a professionally operated ETL company, which allows customers to plan and load analytical data quickly.

2.5 ER DIAGRAM

In the Fig. 2.1 the ER Diagram of Glue Migration is shown where the Task was to Migrate all the data from HIVE the source to Destination GLUE, here there are various pipelines for respective Data sources from HIVE which push data to GLUE. All the pipelines were indexed with Airflow DAGs.

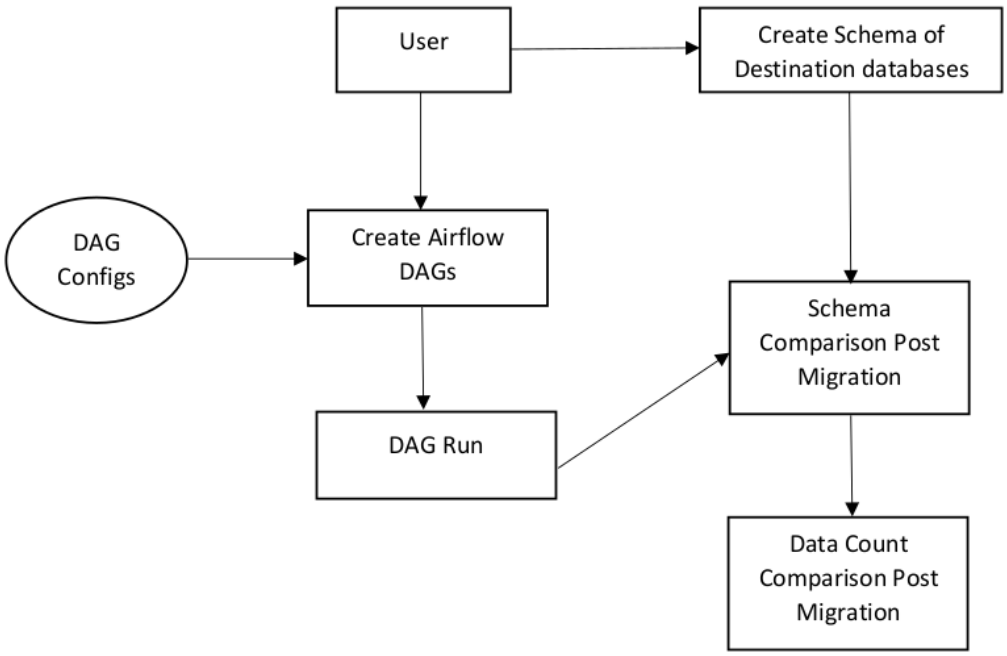

Fig 2.1: ER Diagram of Glue Migration

2.6 IMPLEMENTATION DETAILS

This section contains a brief note of all the technologies used in the development of this project for Frontend and Backend. The Hardware requirements used in developing the Search Application is Laptop or Desktop with RAM of 4GB and above. Below all the Software requirements are mentioned required for developing the Search Application.

2.6.1 Software Interface

Language	:	Python
Platform	:	Databricks, DB Visualizer, Airflow, Putty
Backend	:	Hive, Glue
Databases	:	DynamoDB

- **Databricks**

Databricks is a company founded by Apache Spark's original developers. Building on the University of California's AMPLab project, Berkeley, Databricks designed the Apache Spark, a distributed open source computation platform built on Scala's surface. Databricks provides a web-based platform that offers automated cluster management and notebooks like IPython for Spark working. The organization co-organizes large-scale open online courses about Spark and also runs the biggest Spark Summit conference in addition to developing the Databricks website.

- **DB Visualizer**

To developers, DBAs and analysts DbVisualizer is the universal database program. This is the ultimate solution as all main OSes that access a number of databases will use the same method.

The most relevant databases and JDBC drivers are being checked for DbVisualizer. We introduced support for different database features for more widely used databases in the industry. Runs on Windows, Linux, MacOS and other computers.

- **Airflow**

Apache Airflow is a tool to handle open-source workflows. In October 2014, it started at Airbnb as a solution for managing the increasingly complex processes of the company. The development of Airflow allowed Airbnb to schedule and control its workflows programmatically through the integrated Airflow user interface. Airflow is written in Python and workflows are generated using Python scripts based on its popularity as the de facto programming language for data. Under the "application configuration" concept, airflow is built. Although there are other workflow frameworks for "Application Configurations" using mark-up languages such as XML, developers with Python can import libraries and classes to assist them in building workflows.

- **PuTTy**

PuTTy is a Windows and Linux terminal machine simulator. We have a remote machine text user interface for all of their licensed protocols, including SSH and Telnet. An example of a PuTTY SSH session is shown here. PuTTY is an emulator, serial console, and the framework for network transfer files free and open-source. Support several network protocols, such as SCP, SSH, Telnet, rlogin, and the raw socket. It also has a serial port connection. No official meaning is granted to the word "PuTTY."

PuTTY was first written in Windows for Microsoft, but later moved to many other operating systems. Official ports with working-in-progress ports to classical Mac OS and macOS are available for certain Unix-like systems and unofficial ports such as Symbian, Windows Mobile and Windows Mobile have been contributed.

- **Hive**

Apache Hive is an Apache Hadoop database software warehouse project for data collection and query. Hive offers the SQL-like interface to access data stored in various Hadoop databases and file systems. In order to run SQL applications and queries over distributed data, conventional SQL queries must be implemented in the MapReduce Java API. Hive provides the required SQL abstraction for integration into the underlying Java of SQL-like queries (HiveQL), with no prerequisite for querying the low-grade Java API. Because most data storage systems use query languages based on SQL, Hive supports the portability of SQL systems in Hadoop. Apache Hive is used and developed by others including Netflix and the Financial Industry Regulatory Authority (FINRA) during Facebook's initial growth. Maintained in Amazon's Elastic MapReduce version of the Apache Hive Software branch.

- **Glue**

AWS Glue is a fully managed ETL service which allows customers to easily prepare and load their analysis data. The AWS Management Console helps you to build and execute an ETL job with a few clicks. You just point AWS Glue to your AWS data, and AWS Glue recognizes your data in the AWS Glue Data CatLog and stores the related metadata (e.g. table description and schema). Your data can be scanned, requested and available for ETL immediately after cataloguing. AWS Glue is an ETL service which allows clients to read and load their data for analytics and to extract, transform and load. You can build and run an ETL job in the AWS Management Console with just a few clicks. You just point AWS Glue to your AWS data, and AWS Glue recognizes your data in the AWS Glue Data CatLog and stores the related metadata (e.g. table description and schema).

2.7 METHODOLOGY

This section contains the flow of the GLUE Migration and all the modules included in this application are described in detail. This complete project is divided into different Modules as follows:

i.) Create Airflow DAGS
ii.) Create schema of Destination Database
iii.) Run the Airflow DAG of Migration
iv.) Compare the Schema between Source and Destination Databases
v.) Compare data count between Source and Destination Data Tables

i.) Create Airflow DAG:

The whole process of Migrating data from HIVE to GLUE is long and tough, hence these tasks are automated by using Airflow DAGS, wherein you can streamline all task by creating the DAGS by writing a Python Script which has configurations of that DAG and all its tasks are described with file paths in DAG configs. The development of Airflow allowed Airbnb to schedule and control its workflows programmatically through the integrated Airflow user interface. Airflow is written in Python and workflows are generated using Python scripts based on its popularity as the de facto programming language for data.

ii.) Create schema of Destination Database:

It is required to create schema for destination databases similar to source database's schema, therefore post migration all the data would be aligned accordingly and there would be no mis-migration of data.

iii.) Run the Airflow DAG of Migration

Migration process is performed by the Airflow DAGS, so we need to initiate its run from PuTTY the Unix terminal or from Airflow's UI. The Airflow UI shows all the states of the DAG run and notifies the user when the DAG run successfully completes and when it fails.

iv.) Compare the Schema between Source and Destination Databases

We need to have a comparison between the Source database's schema and destination database's schema, post migration process completes because it is possible that while migration some data might have missed and got failed to migrate hence it can be migrated again.

v.) Compare data count between Source and Destination Data Tables

We need to have a comparison between the Source data count and destination data count, post migration process completes because it is possible that while migration some data might have missed and got failed to migrate hence it can be migrated again.

2.8 SCREENSHOTS

2.8.1 Airflow UI with all DAGS and their information

In the Fig. 2.2 the Airflow DAGs are represented to Migrate all the Data sources from HIVE which push data to GLUE. All the pipelines were indexed with Airflow DAGs.

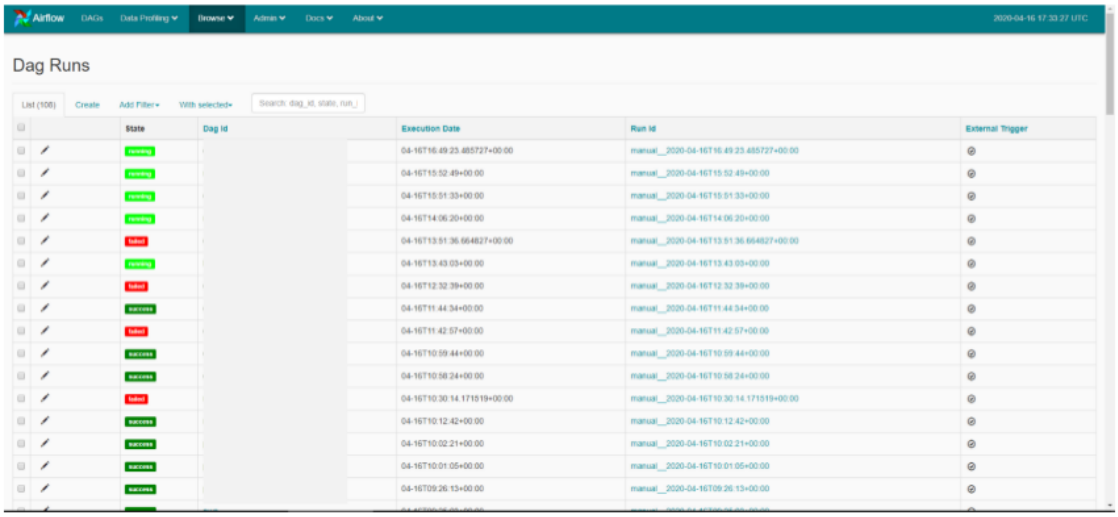

Fig 2.2: Airflow UI with all DAGS and their information

2.8.2 Airflow DAG View

In the Fig. 2.3 the Airflow DAG View is represented where an Airflow Dags is equivalent to a pipeline to Migrate the Data from a specific source from HIVE and push data to GLUE.

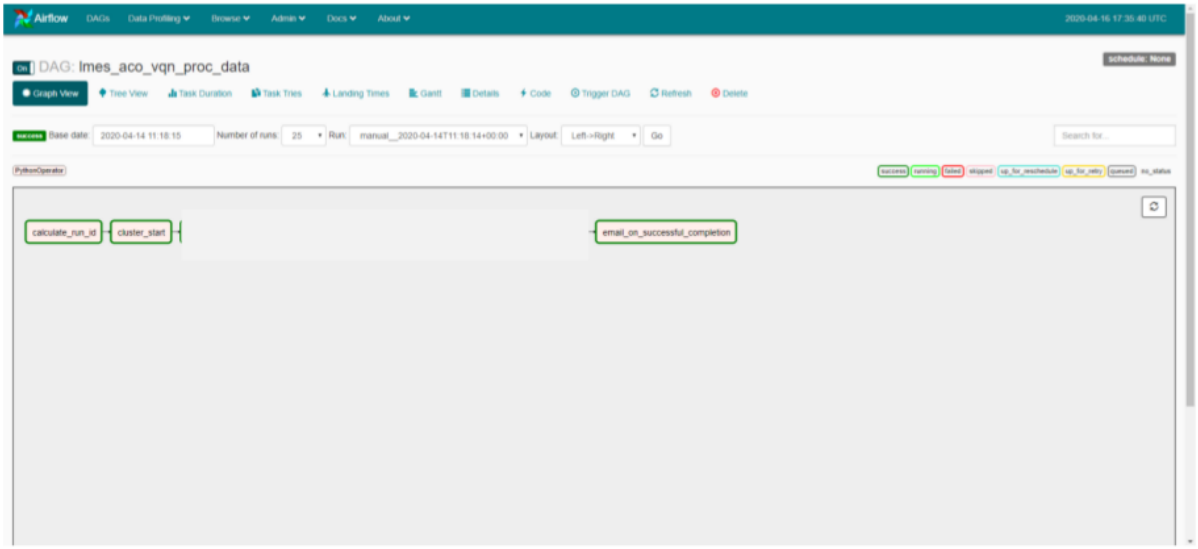

Fig 2.3: Airflow DAG View

CHAPTER 3: COVID-19 APPLICATION

COVID-19 Application is a web application for our pharmaceutical client in that, the application will assist them in planning and supplying their drug production effectively. The application includes all data from Covid-19 areas worldwide and provides a sample of reports on how the production and supply chain of Covid areas can be affected.

3.1 OVERVIEW

In Covid-19 Application, wherein we are creating an application for our Pharmaceutical client, which will help them to effectively plan their medicinal production and supply of same. The application has all data of Covid-19 affected areas from all over the world and gives a projection to the clients with reports which suggests them how their Production and Supply Chain can be affected from Covid affected areas. Raw Data coming from sources i.e. Client datastore or Worldometer is pre-processed using Machine Learning Algorithms. After completing Pre-processing the data files are converted to JSON files. The data files coming from Source post processing converted to JSON which are indexed and stored in Amazon S3 in various S3 Buckets. The data stored in Amazon S3 buckets will be stored in the database i.e. Solr, hence to push this data GraphQL API is used which will index all the data in the Solr. The data stored and indexed on Solr will be displayed on UI as various Data Sets, to push the data from Solr to UI GraphQL API is used. The UI is created using Kepler.gl which is a geospatial analysis tool mainly used for graphically representable data along with HTML, CSS, and React.js framework. The UI is integrated with SSO Single Sign ON with which the Application uses the credentials from Browser and user is not required to Sign In again and again in UI. The UI has the following Modules:

a.) Global Map Page
b.) Count Comparison Page
c.) Summary Page

3.2 PURPOSE

The Covid 19 Application contains all information about Covid-19 regions around the globe, as well as reports to customers about how their production and delivery from Covid areas could affect. In Covid-19, we create a pharmaceutical client application, which will help them effectively plan their medically active production and supply of the same. Automate processes to track the impact of COVID-19 on our Customer's product shipments across Sites, set up algorithms to provide insights through the React App or Report Analytics. Provide a consolidated platform for analytics to assess delays in overall performance. Raw Data coming from sources i.e. Client datastore or Worldometer is pre-processed using Machine Learning Algorithms. After completing Pre-processing the data files are converted to JSON files. The data files coming from Source post processing converted to JSON which are indexed and stored in Amazon S3 in various S3 Buckets. The data stored in Amazon S3 buckets will be stored in the database i.e. Solr, hence to push this data GraphQL API is used which will index all the data in the Solr. The data stored and indexed on Solr will be displayed on UI as various Data Sets, to push the data from Solr to UI GraphQL API is used.

3.3 MOTIVATION

A request from our Pharmaceutical customer to help them effectively plan and supply their medicinal products. The application contains all data on the Covid-19 regions worldwide and gives the customers with reports on how their production and supply chain from Covid areas can be affected. Automate processes to track impact of COVID-19 on our Client’s product shipments across Sites, setup algorithms to derive insights through the React App or from Reports analytics. Provide a consolidated platform to view analytics to assess delays on overall performance. The application has all data of Covid-19 affected areas from all over the world and gives a projection to the clients with reports which suggests them how their Production and Supply Chain can be affected from Covid affected areas.

3.4 OBJECTIVE

The Covid 19 Application provides details on Covid-19 regions worldwide and advises our client about how it can affect their manufacture and distribution from Covid affected areas. We create a web application for our pharmaceutical client, that will help them plan their medically active manufacture and supply. Automate processes to track COVID-19 impacts on site-scale products shipped by our customer, establish algorithms for insight in the reaction application or report analysis. Provide a consolidated analytical platform to evaluate overall performance delays.

Automate processes for tracking the effect of COVID-19 on product shipments across sites of our customers, set up algorithms for analysis through the Reaction application or through analytical reporting. Provide a forum for consolidating research to determine output delays. Raw Data from sources that are i.e. Using machine learning algorithm, consumer data stores or worldometer are pre-processed. The data files are translated to JSON files after pre-processing is done. The data files resulting from the post processing of Source are translated to JSON and stored in various S3 Buckets in Amazon S3. The database will also store the data contained in Amazon S2 bins, i.e. Solr, therefore, uses the GraphQL API data to index all Solr data. Data saved and indexed on Solr will appear as different data sets on the UI to transfer the Solr data from the GraphQL API to the UI. The UI comes from Kepler.gl, a tool used primarily for geospatial info, along with an HTML, CSS and React.js system, which is representable graphically.

The UI is integrated with SSO Single Sign In that makes it impossible for the application to repeatedly sign in the UI with the credentials from the browser. COVID-19 The program is a software application for our pharmaceutical customer that helps them prepare and efficiently produce their medicines. The program contains all data from Covid-19 regions around the world and provides a collection of studies on the potential for effect of the Covid output and supply chain.

3.5 ER DIAGRAM

In the Fig. 3.1 the ER Diagram of Covid-19 Application is shown where the whole flow of the Application is described.

Here, the raw data coming from sources i.e. Client datastore or Worldometer is pre-processed using Machine Learning Algorithms to convert them into JSON files. After completing Pre-processing the data files are converted to JSON files. The data files coming from Source post processing converted to JSON which are indexed and stored in Amazon S3 in various S3 Buckets. The data stored in Amazon S3 buckets will be stored in the database i.e. Solr, hence to push this data GraphQL API is used which will index all the data in the Solr. The data stored and indexed on Solr will be displayed on User Interface as various Data Sets, to push the data from Solr to User Interface GraphQL API is used.

The User Interface is created using Kepler.gl which is a geospatial analysis tool mainly used for graphically representable data along with HTML, CSS, and React.js framework. The User Interface is integrated with SSO Single Sign ON with which the Application uses the credentials from Browser and user is not required to Sign In again and again in User Interface. The User Interface Home Page has various Filters that govern the Global Map View results and there is a Timeline also which decides the time span of results appearing on the Global Map.

The Count Comparison Page has distribution of COVID-19 cases on daily basis across all the countries highlighted by category – Confirmed, Recovered & Deaths in form of Bubble Cluster. Toggle is provided to view a grouped count or split by category.

The User Interface has the following Modules:

a.) Global Map Page
b.) Count Comparison Page
c.) Summary Page

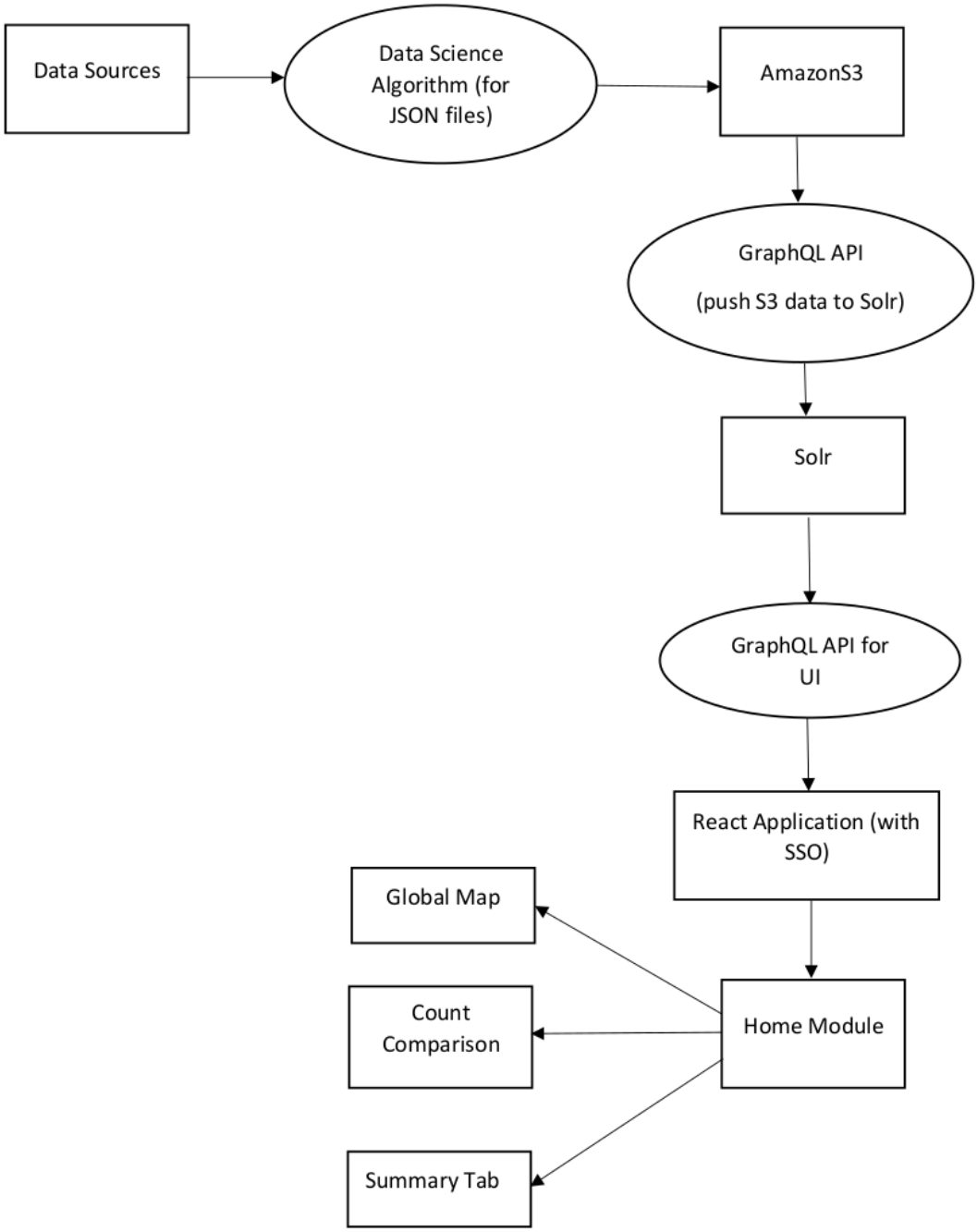

Fig. 3.1: ER Diagram of Covid-19 Application

3.6 IMPLEMENTATION DETAILS

This section contains a brief note of all the technologies used in the development of this project for Frontend and Backend. The Hardware requirements used in developing the Search Application is Laptop or Desktop with RAM of 4GB and above. Below all the Software requirements are mentioned required for developing the Search Application.

3.6.1 Software Interface

Data Sources	:	Worldometer, Client Datastores
Platform	:	Visual Studio
Frontend	:	HTML, CSS, React, Kepler
Backend	:	GraphQL, Python
Databases	:	Amazon S3, Solr

- **Worldometer**

Worldometer is a website of reference that contains counters and statistics on various topics in real time. It is owned and operated by the Dadax data company, which produces income through online publicity. During the COVID-19 pandemic, the website became more popular in early 2020. In March 2020, it was cyber attacked. A few days later, the website was struck by a DDoS attack and the information was wrongly displayed for around 20 minutes on its COVID 19 statistics page. In the case of COVID-19 in Vatican, the hacked site showed a drastic increase that caused fear among some social media users. During the COVID-19 pandemic, the website became more popular in early 2020. A few days later, the website was struck by a DDoS attack and the information was wrongly displayed for around 20 minutes on its COVID 19 statistics page. In the case of COVID-19 in Vatican, the hacked site showed a drastic increase that caused fear among some social media users.

- **GraphQL**

GraphQL is a query language for APIs, and a runtime for your current data to satisfy those queries. GraphQL offers a detailed and comprehensible summary of the data in your API, allows consumers the opportunity to ask for exactly what they need, and nothing more makes APIs simpler to grow over time, and unlocks powerful development tools.

It is an open-sourced search engine that stores data in tokens type in containers. It offers an approach to the creation of web APIs and was compared with REST and other architectures for web services. It allows clients to specify the structure from the server, therefore preventing the return of overly large quantities of data, but this has consequences for how efficient web caching of query results can be. The versatility and richness of the query language also adds complexity for simple APIs.

- **Python**

It is the high level of dynamic semantism, interpreted, and oriented programming language. Its high degree of compatibility is built into data structures in conjunction with dynamic typing and scripting. The plain, easy-to-learn syntax of Python underlines readability and thus reduces the maintenance costs. The Python interpreter and its broad standard library can be distributed free of charge in source or binary form for all major platforms. Python is a programming language of many paradigms. Objective-driven programming and organized programming are completely embraced, and many of its features support functional programming and aspect-oriented programming. Several other paradigms, including contract architecture and logic programming, are supported by extensions.For memory management, Python uses dynamic typing, and a combination of reference counting with cycle detection of waste. This has also a late binding dynamic name resolution, which connects the process and variable names while running the program.

- **HTML**

The basic language for mark-ups for the documents intended to appear on a web-browser is the hypertext mark-up language (HTML). Technologies like the cascade of style sheets (CSS) and scripting languages like JavaScript can be used to help this. HTML documents are downloaded from a web server or local storage from Web browsers and translated to the web pages of multimedia. HTML explains the web page layout in a comprehensive manner and the documents originally included. Building blocks of HTML pages are HTML elements. The made page can be incorporated with HTML builds, pictures and other objects such as interactive forms. For text, including headings, sections, lists, links, quotations, and other things HTML offers a way to produce organized documents by defining structural semanticity. HTML elements are labelled, written with angle brackets. Tags like and include material directly in the tab. Sub-elements may also include other tags, such as surround and provide information on document texts. The HTML tags are not viewed by the browsers but are used by them to view the page content.

- **CSS**

Cascading Style Sheets (CSS) is a style plate term used to characterize the application of a markup document such as HTML. This separation will increase usability of contents, provide greater flexibility and control over the presentation characteristics, allow multi-web sites to share the formatted content by deciding which one is the basis. The CSS is a key technology in the World Wide Web alongside HTML and JavaScript. CSS for separating presentation and contents including templates, colors and font. Separation of formatting and contents also allows to display a single markup page in different styles for different rendering methods , e.g. on-screen, in print, by voice, or on Braille- based tactile tools. If content is viewed on a mobile device, CSS also has alternative formatting guidelines. The name cascading derives from the priority scheme to decide the rule of style if more than one rule fits a particular feature. This goal is predictable in cascading.

- **React**

Reaction is a JavaScript library used for designing user interfaces. React (also known as React.js or ReactJS). Facebook and a consortium of developers and businesses are sustaining it. In developing single-page or mobile applications, react may serve as a base.

However, React is restricted to making data to the DOM, which includes the use of additional libraries for state management and routings. React typically needs to build React applications. Redux and React Router are examples of such libraries respectively. React component that accepts a property greeting is Greeter function. The ReactDOM.render method then makes our Greeter component with Id myReactApp inside the DOM element.

- **Kepler**

Kepler.gl is a part that has a Redux relation. You can incorporate kepler.gl with RedX to manage its status into your app. Kepler.gl reducer's basic application is simple. To take advantage of this, however, it is advisable to know basic things:

- React
- Redux state container
- React Redux connect

You only need to add the Kepler.gl UI portion and install the Kepler.gl reducer to continue with Kepler.gl. This package also contains actions, schema managers and a set of map data storage utilities for the user to have full access to all the functionalities of kepler.gl. To help data scientists work more efficiently, we have incorporated kepler.gl into a range of commonly used data processing tools, including Jupyter Notebook. Jupyter Notebook is a popular open source web application used to build and distribute live code, equations, visualizations, and text documents, primarily used by data scientists to perform data analysis and distribute results.

- **Visual Studio**

Microsoft Visual Studio is a Microsoft-based integrated software (IDE) environment. It is used for computer software and for web applications and mobile applications. Web applications. Visual Studio uses the Windows API, Windows Forms, the Windows Presentation Foundation, the Windows Store, and Microsoft Silverlight software development platforms in the field. It can generate both native and controlled code. The Visual Studio comprises an IntelliSense code editor (component for the completion of code) and a code reactor. The built-in debugger acts as both a source and a computer debugger. The application profiler, the Interface designer, the web designer, the class designer, and the database schema designer are some of the integrated tools. It accepts plug-ins that improve functions on nearly every level — including adding support for source control systems (such as Subversion and Git) and adding new tool sets for domain-specific languages or toolkits, such as editors and visual designers for other aspects of the development of software.

- **Amazon S3**

Amazon Simple Storage Service provides storage of objects through cloud services. It is also known as Amazon S3 which is an Amazon Web Service. Amazon S3 uses the global e-commerce network with the same flexible storage system that Amazon.com does. The Amazon S3 can be used for storing items of any kind, such as internet storage, security and recovery, disaster recovery, collections of data, analytics data lakes, and hybrid cloud storage. AWS launched Amazon S3 on March 14, 2006 in the United States and in November 2007 in Europe. Amazon's easy storage service (Amazon S3) provides leading scalability, availability of content, protection, and efficiency in an artifact storage facility. This helps consumers of all sizes and industries to store and secure any amount of data in a variety of uses, including websites, mobiles, backups and reconstruction, libraries, business apps, IoT devices and Big Data Analysis.

Amazon S3 offers easy-to-use management functions, allowing you to arrange your data to comply with your particular sector, organization and conformity criteria and configure completed access control systems.

- **Solr**

  Solr (pronounced "solar") is an Apache Lucene project's open-source corporate search engine written in Java. Highlights, facet-level search, real-time indexing, dynamic clustering, data base integration, NoSqL functionality, as well as comprehensive handling of documents (for example Word or PDF) are all functionality of the business. Solr is optimized for scalability and fault tolerance to provide distributed search and index replication. Solr has an active developer community with regular releases and is widely used for corporate search and analytics. The Solr search server is operating autonomously. It utilizes the full-text indexing and search library for Lucene Java, and it has HTTP/XML and JSON-like REST APIs that support it in the most common programming languages. The external Solr software enables it to be customized without Java coding to several applications and has a plugin architecture that facilitates more customization.

3.7 METHODOLOGY

This section contains the flow of the COVID-19 Application and all the modules included in the application are described in detail. This project is divided into different Modules as follows:

i.) Conversion of Source Data Files to JSON
ii.) Store the JSON data files in Amazon S3
iii.) Use GraphQL API to push data to Solr from S3
iv.) Push Solr data to UI using GraphQL API
v.) React Application using Kepler
    d.) Global Map Page
    e.) Count Comparison Page
    f.) Summary Page

i.) Conversion of Source Data Files to JSON

Raw Data coming from sources i.e. Client datastore or Worldometer is pre-processed using Machine Learning Algorithms. After completing Pre-processing the data files are converted to JSON files.

ii.) Store the JSON data files in Amazon S3

The data files coming from Source post processing converted to JSON which are indexed and stored in Amazon S3 in various S3 Buckets.

iii.) Use GraphQL API to push data to Solr from S3

The data stored in Amazon S3 buckets will be stored in the database i.e. Solr, hence to push this data GraphQL API is used which will index all the data in the Solr.

iv.) Push Solr data to UI using GraphQL API

The data stored and indexed on Solr will be displayed on UI as various Data Sets, to push the data from Solr to UI GraphQL API is used.

v.) React Application using Kepler

The UI is created using Kepler.gl which is a geospatial analysis tool mainly used for graphically representable data along with HTML, CSS, and React.js framework. The UI is integrated with SSO Single Sign ON with which the Application uses the credentials from Browser and user is not required to Sign In again and again in UI.

The UI has the following Modules:

a. Global Map Module:

It is the Home Page of the UI which consists of Filters that can be applied on the Global Map regarding New Infection cases, New Deaths, New Recovered cases. It also consists a Timeline Bar through which a time range can be selected, and the Infection is shown in form of Bubbles on the UI. Provides a progressive view of the covid-19 infections spreading with time and a quick glimpse of Client sites that are impacted. Health Check view left of the Layers section will be provided for user to update the health based on the analytics.

b. Count Comparison Module:

This Page has distribution of COVID-19 cases on daily basis across all the countries highlighted by category – Confirmed, Recovered & Deaths in form of Bubble Cluster. Toggle is provided to view a grouped count or split by category.

c. Summary Module:

This Page consists of the Summary related to all countries data regarding the Confirmed, Recovered & Deaths and has a bar plot for comparison.

3.8 SCREENSHOTS

3.8.1 Solr Displaying Data as per query

In the Fig 3.2 A Query is executed on Solr with a fq (filter query) is applied. Solr is used for data storage where user can not only store the data but has various and easy approaches to Query the data. The Data stored in Solr is stored into Collections and Aliases.

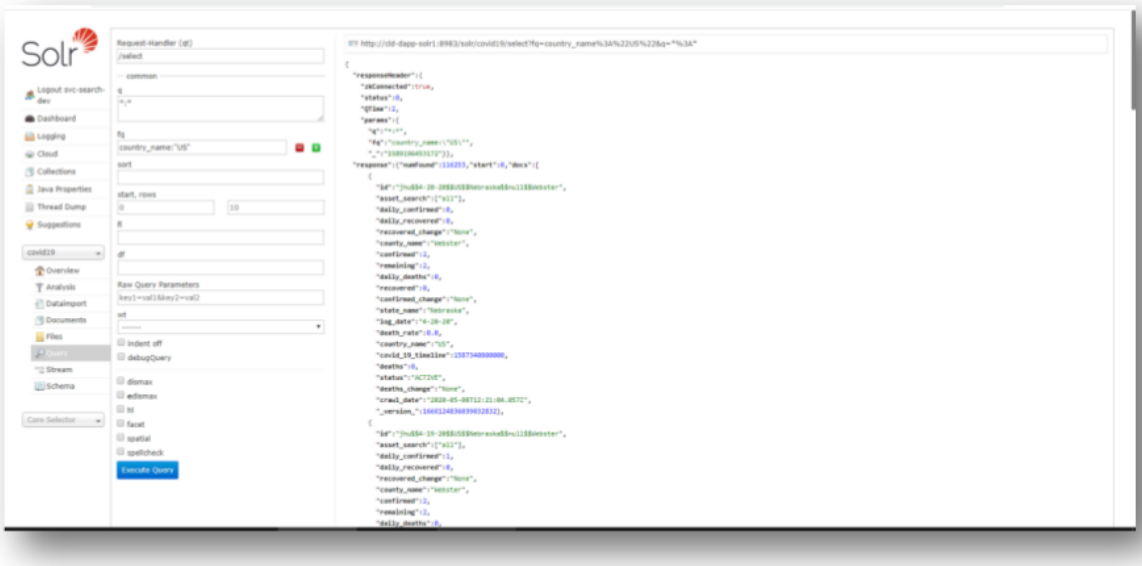

Fig 3.2 Solr Displaying Data as per query

3.8.2 Amazon S3 View

In the Fig. 1.3 the User Interface of Amazon S3 is shown, which stores data in form of Buckets. It not only provides feature of data storage in buckets but also allows user to perform general Operations on it.

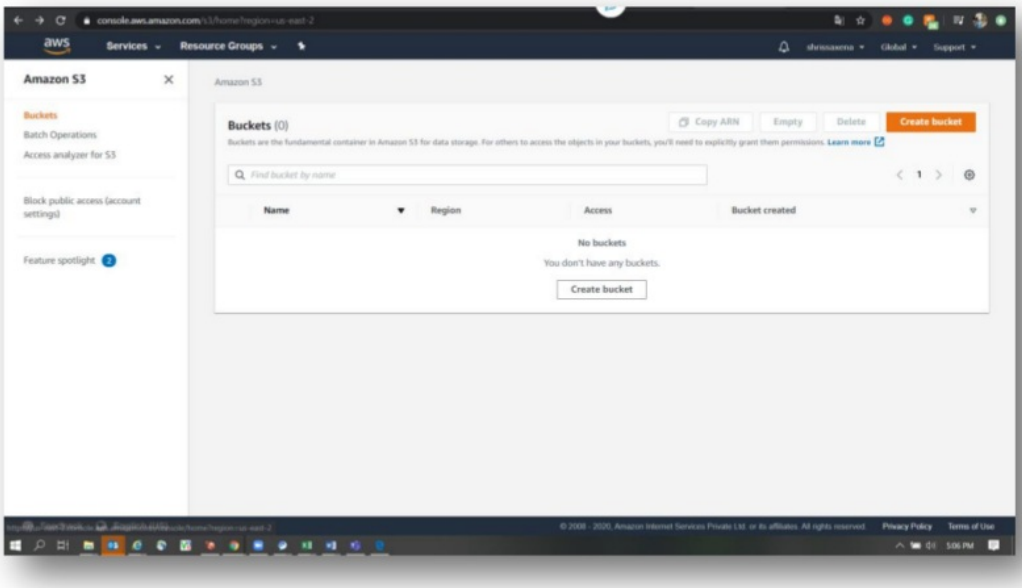

Fig 3.3 Amazon S3 view

3.8.3 Kepler Demo View

In the Fig. 3.4 the Kepler Demo View is shown which is implemented in the User Interface.

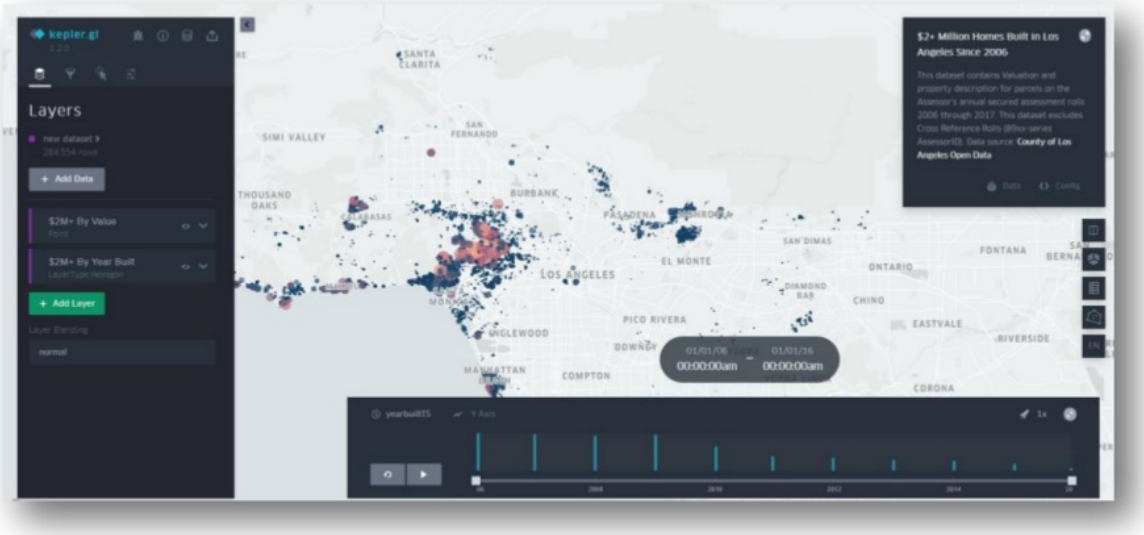

Fig 3.4 Kepler Demo View

REFRENCES

1) Sandeep Nair, *Mastering Apache Solr,* Chintan Mehta, 2018

2) Arun Kumar, *The Ultimate Putty Guide*, Vijendra Sinha, 2019

3) Apache HIVE from- Wikipedia

4) Guidance on Amazon Glue from- aws.amazon.com

5) Guidance on DB Visualizer from- www.dbvis.com

6) Apache Airflow from- Wikipedia

7) Guidance on Kepler from- www.kepler.gl.com

report

ORIGINALITY REPORT

32%	28%	5%	30%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to DIT university Student Paper	15%
2	www.dituniversity.edu.in Internet Source	4%
3	www.ijser.org Internet Source	4%
4	Submitted to ABV-Indian Institute of Information Technology and Management Gwalior Student Paper	3%
5	88306.com Internet Source	2%
6	psasir.upm.edu.my Internet Source	2%
7	docplayer.net Internet Source	2%

Exclude quotes	Off	Exclude matches	Off

Exclude bibliography On