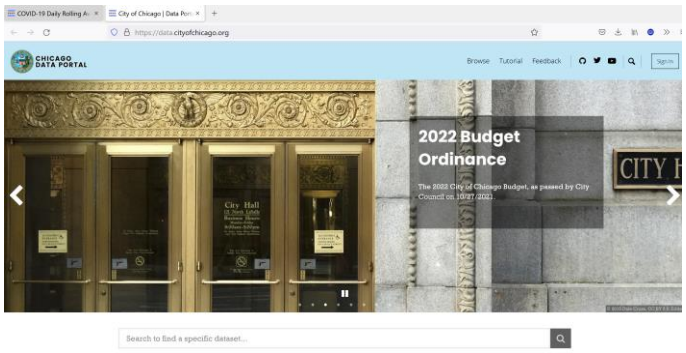
Assignment #4

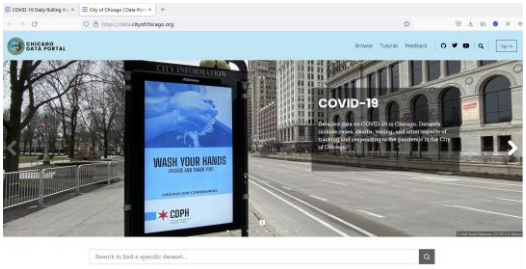
Assignment Submission:

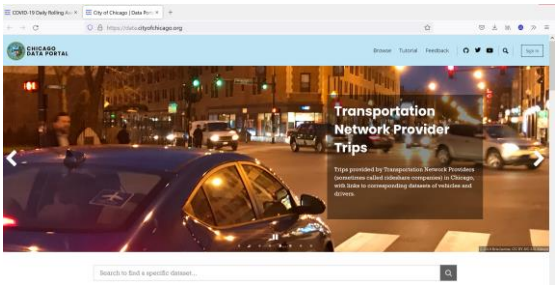
Submit your assignment on Blackboard as a SINGLE ZIP file that has

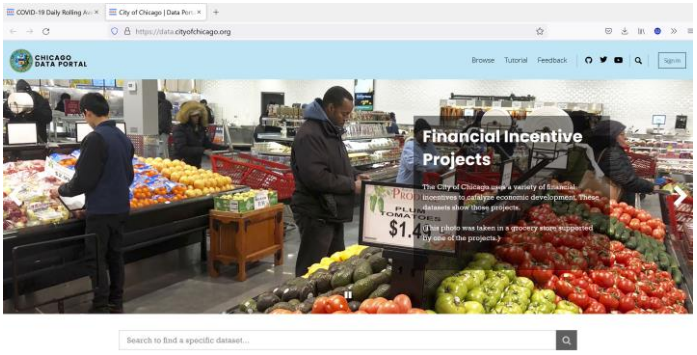
1. PDF document for the requirements listed below.
2. All source code created and built
3. Sample of datasets reviewed and used
4. Video recording of the live-run demo for the requirements listed below.

Requirements:









After you review and analyze the requirements specification document for **Chicago Business Intelligence for Strategic Planning** project, provide your answers for the requirements listed below:

Requirement 1: Create a table that shows the dataset name and the URL listed on the City of Chicago data portal and the requirements in the requirements specification document of the Chicago Business Intelligence for Strategic Planning that need that data source.

Requirement 2: Does every data source (dataset) have all attributes needed for every report/query required for Chicago Business Intelligence

for Strategic Planning? For example, does Building Permits (https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.html) dataset have Zip Code? does the Health Stat dataset (https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Selected-public-health-in/iqnk-2tcu/data) and (https://data.cityofchicago.org/resource/iqnk-2tcu.json) to find the unemployment and poverty level data for the different community areas have the Zip Code?

Requirement 3: What are the tables and their attributes, data types that you created for the Data Lake. What is the database engine that you used for your Data Lake? Explain in detail if you need to create a table for every dataset and if you need to create a new table to store results of merging data from different datasets.

Requirement 4: Create a table that shows the dataset name and the URL listed and whether it has Neighborhood Names, Community Areas, and Zip Codes.

Requirement 5: Explain how you cross-reference Neighborhood Names, Community Areas, and Zip Codes

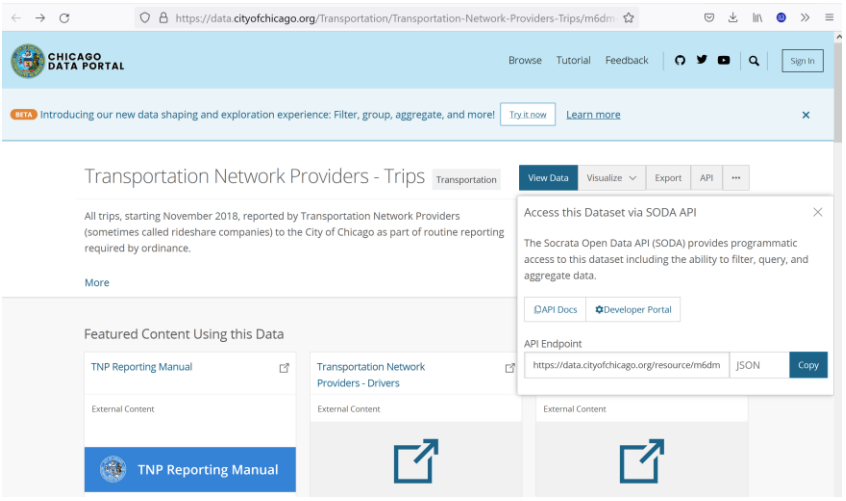
Requirement 6: Explain why there are two data sources for the transportation: (https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew) and (https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips/m6dm-c72p)

Requirement 7: List all reports that are needed to meet the requirements for the Chicago Business Intelligence for Strategic Planning project. For example, a report is needed to provide all information of the taxi trips from O'Hare and Midway airports to the different zip codes.

Requirement 8: Create a table that has the name of every report you identified in the prior requirement, and document for every report the following:

1. Report Name

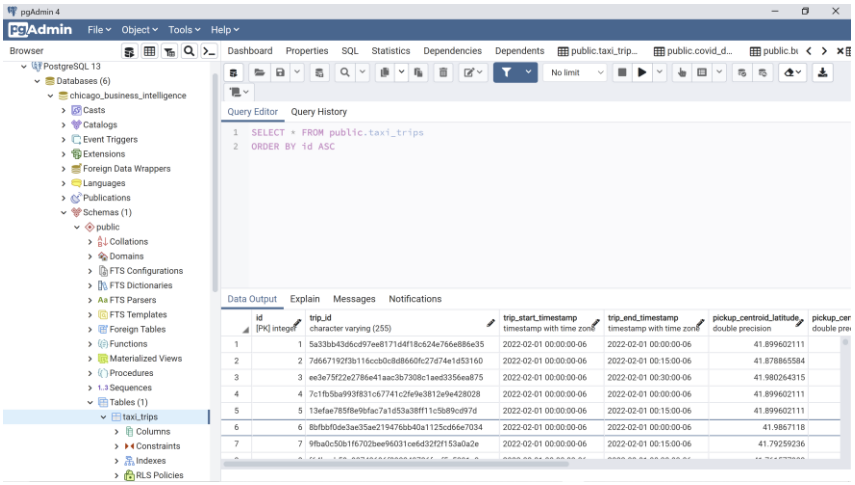
2. Data Source. For example, Transportation/Trips is (https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips/m6dm-c72p)
3. Data API – End Point. For example, Transportation/Trips is (https://data.cityofchicago.org/resource/m6dm-c72p.json)



4. List the attribute names and data types that you will use from every dataset
5. If there are multiple datasets used to construct certain table, list all of these. For example, the taxi trips has the two data sources the taxi medallion (https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew/) and the ride shares (https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips/m6dm-c72p/)

6. Document the data preparation/preprocessing steps that are needed to deal with messy/dirty/missing values for every attribute in every dataset. For example, for the taxi trips, if we have a record that is missing the values for any of the attributes, like pick-up and drop-off times for example, we will not insert that record into the taxi trips table in the data lake.

7. List the attribute names and data types that you will create for the table representing every dataset in your data lake on the database engine. For example, the following figure shows Postgres database engine that has the chicago_business_intelligence data lake that hosts the following table:


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
PostgreSQL 13
Databases (6)
chicago_business_intelligence
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (1)
taxi_trips
Columns
Constraints
Indexes
RLS Policies
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
public.taxi_trip...
public.covid_d...
No limit
Query Editor
Query History
SELECT * FROM public.taxi_trips
ORDER BY id ASC
Data Output
Explain
Messages
Notifications
id
[PK] integer
trip_id
character varying (255)
trip_start_timestamp
timestamp with time zone
trip_end_timestamp
timestamp with time zone
pickup_centroid_latitude
double precision
1 1 5a33bb43d6cd97ee8171d4f18c624e766e886e35 2022-02-01 00:00:00-06 2022-02-01 00:00:00-06 41.899602111
2 2 7d66719 2f3b116ccb0c8d8660fc27d74e1d53160 2022-02-01 00:00:00-06 2022-02-01 00:15:00-06 41.87886558 4
3 3 ee3e75f22e2786e41aac3b7308c1aed3356ea875 2022-02-01 00:00:00-06 2022-02-01 00:30:00-06 41.980264315
4 4 7c1fb5ba993f831c67741c2fe9e3812e9e428028 2022-02-01 00:00:00-06 2022-02-01 00:00:00-06 41.899602111
5 5 13efae785f8e9bfac7a1d53a38ff11c5b89cd97d 2022-02-01 00:00:00-06 2022-02-01 00:15:00-06 41.899602111
6 6 8bfbbf0de3ae35ae219476bb40a1125cd66e7034 2022-02-01 00:00:00-06 2022-02-01 00:00:00-06 41.9867118
7 7 9fba0c50b1f6702bee96031ce6d32f2f153a0a2e 2022-02-01 00:00:00-06 2022-02-01 00:15:00-06 41.79259236

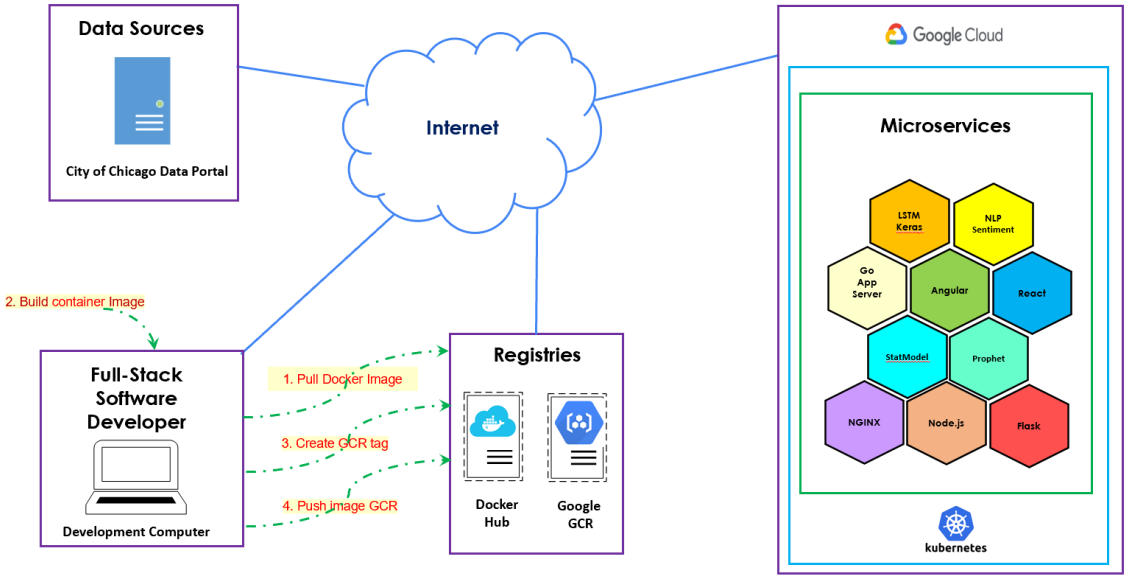
Requirement 9: Use the SODA API to inspect and retrieve 2000 records by typing the following URL (https://data.cityofchicago.org/resource/wrvz-psew.json?$limit=2000) in the FireFox/Chrome browser.

> **Note: It might take the data portal server up to 5 minutes to return the records.**

Requirement 10: Use the SODA API to inspect and retrieve 2000 records by typing the following URL (https://data.cityofchicago.org/resource/m6dm-c72p.json?$limit=2000) in the FireFox/Chrome browser.

Requirement 11: Consider the following architecture and design diagram to build the cloud-native microservice for the Chicago Business Intelligence for Strategic Planning project. Identify and document the steps and workflows for data collection, data preprocessing, and name the microservices needed for deployment; your documentation and annotation of the workflows and steps should be similar to the annotation that you see on the diagram below.

Architecture & Design for Cloud-Native Microservices



Requirement 12: List the names of the microservices and their purposes that you decided are needed to implement the Chicago Business Intelligence for Strategic Planning project. For example, you might state that you need to pull Postgres image from Docker to be your database engine.

Requirement 13: Compare and contrast your design considering the following options:

1. Use the personal development computer native operating system (OS) to build, deploy, and run loosely-coupled programs (microservices)
2. Use Docker build, deploy, and run containers for the difference microservices on the development computer native operating system (OS)
3. Use Google Cloud to build, deploy, and run the needed microservices

Requirement 14: Implement all required **Back-End** microservices using any of the modern programming languages: Golang, Python, Java, JavaScript, etc.

Deliverables:

1. You must submit a single ZIP file that has the subdirectories: **src** and **doc**
2. The PDF document that has the answers for the requirements 1 through 13, listed above, should be stored in the **doc** directory
3. Complete implementation for **Requirement 14**; **Back-End** microservices
4. Source code for the microservices/programs you created should be stored in the **src** directory
5. A 5-minutes video demo of your live-run should be stored in the **doc** directory
6. Readme file that has the list of steps to build and run your program should be stored in the **doc** directory