**Subject: INFO7390: Advance Data science and Architecture**

**Assignment 1 – Design and Implementation**

**Team 11**

**Team members:**

**Vaishali Lambe**

**Rijuta Wagh**

**Ankur Muley**

**Assignment 1**

**Problem 1: Data wrangling Edgar data from text files (50 points)**

**Part 1:**

1. Created edgar\_config.py file which contains CIK and Accession numbers
2. First tested that for given user inputs CIK and accession number, able to populate url for IBM
3. Parsed the files for 10q filings for IBM
4. Extracted tables from filings and saved into .csv files

**Part 2:**

1. Created a edgar\_config.py file which contains google CIK and accession number
2. Also, for AWS need to provide below content

cik = ""

accession = ""

aws\_access\_key\_id = ""

aws\_secret\_access\_key = ""

aws\_upload\_bucket\_name = ""

[ in this case, cik is the CIK of the company, e.g. "51143", accession is the accession number for the document to retrieve, e.g. "0000051143-13-000007"

The aws variables are the access key ID, secret access key and bucket name for file uploads.

All variables should be strings.]

1. Populated urls as per user inputs
2. Parsed to see whether it is 10-K or 10-Q filing
3. Extracted all tables from filings
4. Run below command:

docker run -v /PATH/TO/edgar\_config.py:/home/ds/edgar\_config.py edgar-s3

Replace "/PATH/TO/edgar\_config.py " with the real path to the "edgar\_config.py"

Docker hub link:

<https://hub.docker.com/r/vaishalilambe/team11_edgar/>

<https://hub.docker.com/r/vaishalilambe/team11_edgar_s3/>

**Logging:**

1. If CIK is None or CIK is empty value, log an entry to log file with error message
2. If Accession is None or CIK is empty value, log an entry to log file with error message
3. If aws\_access\_key\_id is None or empty value, log an entry to log file with error message
4. If aws\_secret\_access\_key is None or empty value, log an entry to log file with error message
5. If aws\_upload\_bucket\_name id None or empty value, log an entry to log file with error message
6. If created URL is not accessible log an entry to return error status code
7. Log entries based or whether table is consistent or inconsistent

**Problem 2: Missing Data Analysis and Visualization (50 points)**

**Part 1:**

1. Created edgar\_config.py file which takes year=”” as input
2. Program will find the first day of every month data for that particular year
3. Program will automatically generate url for extracting .zip files
4. Logged the actions performed through program
5. Handled missing data for columns 'cik', 'code', 'filesize', 'idx', 'find', 'crawler' and replaced it with zero, the corresponding action is also logged
6. Created dataframe to display the metrics
7. Cleaned data and replaced size with mean value
8. Handled outlier

**Part 2:**

1. Created config file edgar\_logs\_config.py with below content:

    year = ""

aws\_access\_key\_id = ""

aws\_secret\_access\_key = ""

aws\_upload\_bucket\_name = ""

1. Wrote to program to upload logfile to s3 bucket
2. Run below command:

docker run -v PATH/TO/edgar\_logs\_config.py:/home/ds/edgar\_logs\_config.py edgar-logs-s3

Replace "/PATH/TO/edgar\_logs\_config.py " with the real path to the "edgar\_logs\_config.py"

**Visualisation:**

1. Analysed data for 2003
2. With the help of Tableau created dashboard which contains to graphs:

Graph: Browser Vs Size: Which shows Windows browser is used more around 3000M users

Graph: noagent Vs Date/Time: Which gives information of how many users come thorugh agent at specific time and date

Docker hub link:

<https://hub.docker.com/r/vaishalilambe/team11_edgar_logs_s3/>

**Logging:**

1. If year is None or empty, log an entry with error message
2. If aws\_access\_key\_id is None or empty value, log an entry to log file with error message
3. If aws\_secret\_access\_key is None or empty value, log an entry to log file with error message
4. If aws\_upload\_bucket\_name id None or empty value, log an entry to log file with error message
5. If created URL is not accessible log an entry to return error status code
6. Check where year exists on edgar website or not and decide whether entered year is valid year or not and display corresponding error messages
7. Check where .zip is already downloaded or not and display corresponding message

**Submission:**

You will find two folders for each problem

Problem 1:

One contains dockerfile and edgar.py without s3 and aws code(edgar-image)

One contains dockerfile and edgar.py with s3 and aws code(edgar-image-s3)

Problem 2

One contains dockerfile and edgar-logs.py without s3 and aws code(edgar-logs-image)

One contains dockerfile and edgar-logs.py with s3 and aws code(edgar0logs-image-s3)