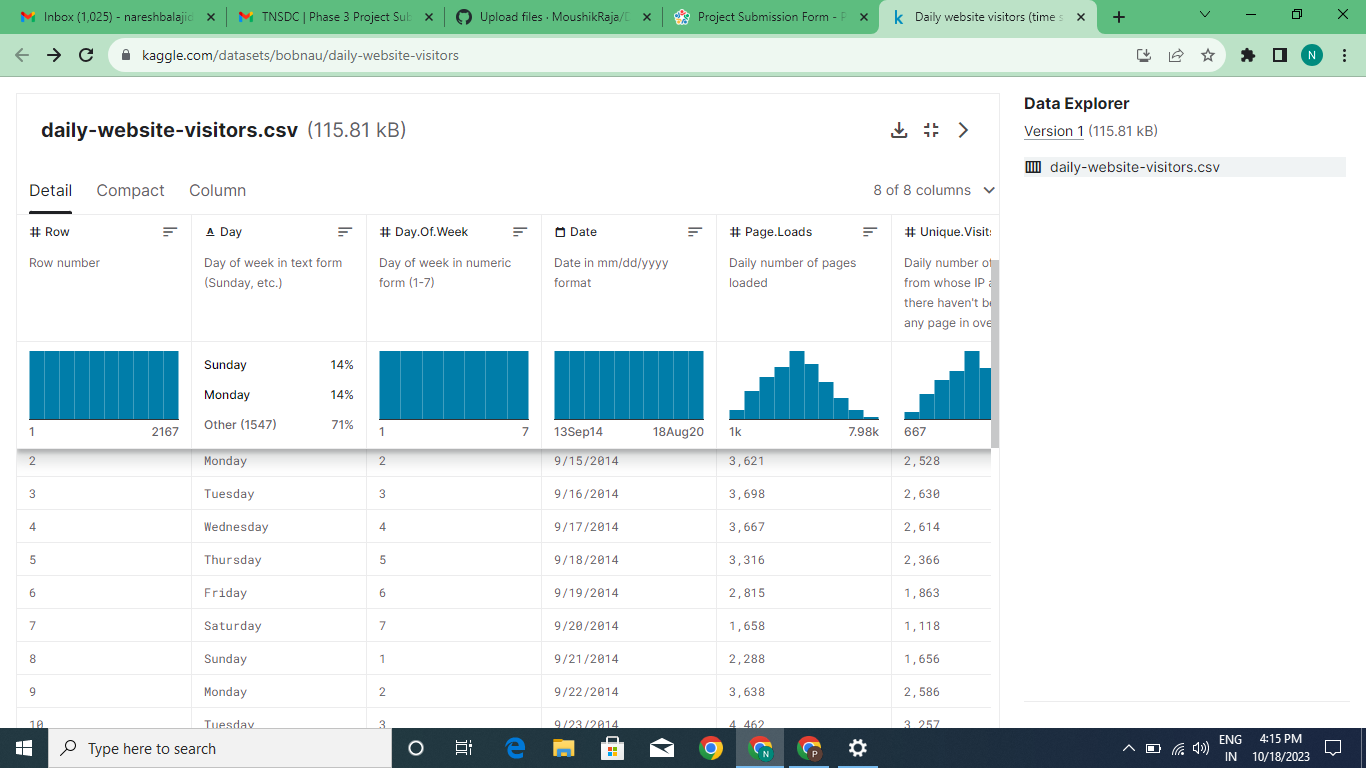
PHASE-3

**Development Part 1**

In this part you will begin building your project by loading and preprocessing the dataset. Start building the website traffic analysis using IBM Cognos for visualization. Define the objectives of the analysis and load website traffic data from the source shared. Process and clean the collected data to ensure its accuracy and consistency.

**Dataset Link:**[**https://www.kaggle.com/datasets/bobnau/daily-website-visitors**](https://www.kaggle.com/datasets/bobnau/daily-website-visitors)



The website traffic analysis of a website can be done under various specific method, such as

1.Data Sourcing

Collecting the data from the dataset link and getting the details about the data of the project from the given source.

2.Data loading

Processing the complete data collected from the datalink and resourcing the source which is collected and analysis in the data sourcing.

3.Data Preprocessing

Data cleaning and preprocessing are critical to ensure the accuracy and consistency of your analysis. This step include:

* Handling missing data: Remove or impute missing values.
* Data transformation: Convert data types, standardize date formats, and handle outliers.
* Data validation: Check for duplicates and inconsistencies in the data.
* Data integration: Data from multiple sources, integrate it to create a unified dataset.

4.Creating data models

In IBM Cognos, need to create data models to represent the structure of your data. These models will help in building reports and dashboards. Also need to define relationships between different data tables.

5.Data Visualization

Utilizing IBM Cognos' reporting and visualization capabilities to create meaningful dashboards and reports.

6.Analysis and Insights

Perform the analysis the visualizations are extract insights from the data.

For Analyzing the dataset

INPUT

import pandas as pd  
  
FILE\_LOCATION = '/kaggle/input/daily-website-visitors/daily-website-visitors.csv'  
  
whole\_dataset = pd.read\_csv(FILE\_LOCATION,   
 index\_col='Date',  
 thousands=',')  
whole\_dataset.index = pd.to\_datetime(whole\_dataset.index)  
whole\_dataset

OUTPUT:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Row | Day | Day.Of.Week | Page.Loads | Unique. Visits | First.Time.Visits | Returning. Visits |
| Date |  |  |  |  |  |  |  |
| 2014-09-14 | 1 | Sunday | 1 | 2146 | 1582 | 1430 | 152 |
| 2014-09-15 | 2 | Monday | 2 | 3621 | 2528 | 2297 | 231 |
| 2014-09-16 | 3 | Tuesday | 3 | 3698 | 2630 | 2352 | 278 |
| 2014-09-17 | 4 | Wednesday | 4 | 3667 | 2614 | 2327 | 287 |
| 2014-09-18 | 5 | Thursday | 5 | 3316 | 2366 | 2130 | 236 |
|  |  |  |  |  |  |  |  |
| 2020-08-15 | 2163 | Saturday | 7 | 2221 | 1696 | 1373 | 323 |
| 2020-08-16 | 2164 | Sunday | 1 | 2724 | 2037 | 1686 | 351 |
| 2020-08-17 | 2165 | Monday | 2 | 3456 | 2638 | 2181 | 457 |
| 2020-08-18 | 2166 | Tuesday | 3 | 3581 | 2683 | 2184 | 499 |
| 2020-08-19 | 2167 | Wednesday | 4 | 2064 | 1564 | 1297 | 267 |

FOR PREPERARING OF DATASET

INPUT:

target\_column = whole\_dataset['Returning.Visits']  
target\_column:

OUTPUT:

Date  
2014-09-14 152  
2014-09-15 231  
2014-09-16 278  
2014-09-17 287  
2014-09-18 236  
 ...   
2020-08-15 323  
2020-08-16 351  
2020-08-17 457  
2020-08-18 499  
2020-08-19 267  
Name: Returning.Visits, Length: 2167, dtype: int64

VISUALIZATION OF VISITS:

INPUT:

dataset\_by\_day = unbatched\_train\_dataset.groupby(by=['Day'])  
dataset\_by\_day['Returning.Visits'].mean()

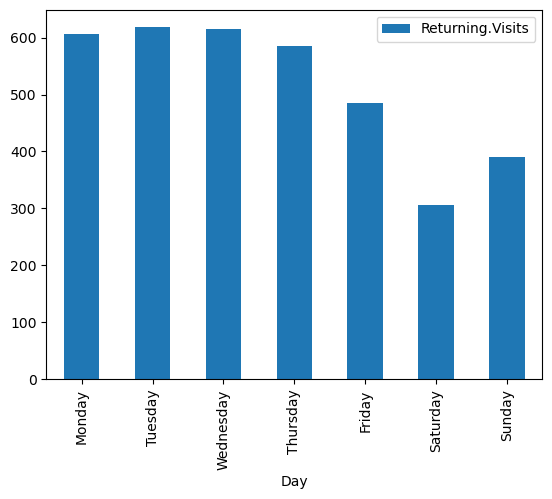
OUTPUT:

Day  
Friday 484.697842  
Monday 606.512545  
Saturday 306.071942  
Sunday 390.573477  
Thursday 584.627240  
Tuesday 617.888889  
Wednesday 614.369176  
Name: Returning.Visits, dtype: float64

INPUT:

DAYS\_OF\_WEEK = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']  
pd.DataFrame(dataset\_by\_day['Returning.Visits'].mean()).loc[DAYS\_OF\_WEEK].plot(kind='bar')

OUTPUT:



INPUT:

dataset\_by\_day['Returning.Visits'].hist(legend=True, alpha=0.5)  
plt.show()

