A TIME SERIES ANALISIS OF ELECTRIC PRODUCTION DATASET

TEAM-6

1	RAMADAGULA KATHIKEYA HARSHITHA	**22A81A05I4**
2	VUMMIDI RAHUL VENKATA SAI ABHIRAM	**22A81A05I5**
3	RAMBHA RASHMITHA LEKA	**22A81A05I6**
4	RAPARLA PRASANNA KUMAR	**22A81A05I7**
5	SALI NAVYA SRI	**22A81A05I8**
6	SAMUDRALA S S N V R DURGA MAHATHI	**22A81A05I9**
7	SATTI BHANU PRAKASH REDDY	**22A81A05J0**
8	SEELABOINA DURGA	**22A81A05J1**
9	TEKI VIJAYARATNAM	**22A81A05J2**
10	TUMMALAPALLI MANIKANTA VINAY	**22A81A05J3**

T

Project Title: TIME SERIES ANALYSIS ON ELECTRIC PRODUCTION DATA SET

Category	TEAM 6
Introduction	"Embark on a journey through the intricate realm of time series analysis with Python, where we unravel the mysteries of temporal data using cutting-edge techniques and libraries." "Welcome to the forefront of data science as we delve into the dynamic domain of time series analysis using advanced Python methodologies, unlocking insights from sequential data streams." "Step into the realm where past, present, and future converge as we harness the power of Python's advanced tools to dissect, model, and forecast time series data, navigating through trends, seasonality, and anomalies with precision."
Data Collection	Web Scraping. API Integration Data Downloads Database Queries
Methodology	During the analysis, I utilized exploratory data analysis to understand the dataset's structure and patterns. Then, I applied statistical tests like ADF for stationarity and modeling techniques such as ARIMA and LSTM for forecasting. Finally, I evaluated model performance using metrics like MAE and visualized results to communicate insights effectively.
	Navigating the dynamic nature of the dataset and selecting

Trainer Name: Er. M. Rajesh kannan. ME.,

	suitable models amid uncertainty posed significant challenges during implementation.
Data Analysis	The data revealed a clear seasonal trend, with distinct patterns recurring over specific time intervals, indicating potential opportunities for leveraging seasonality in forecasting. Additionally, a gradual upward trend in the long-term suggests underlying growth or systemic changes that could impact future predictions and decision-making
	The results of the analysis were validated through cross-validation, comparing forecasted values against actual observations using various metrics to ensure model accuracy and reliability.
Results	The analysis revealed significant seasonal trends and a gradual long-term upward trend, indicating potential forecasting opportunities and underlying systemic changes.
Conclusion	In conclusion, the analysis demonstrates the presence of distinct seasonal patterns and a gradual upward trend in the dataset, suggesting the need for robust forecasting methods to capitalize on opportunities and adapt to evolving trends effectively. This understanding enhances decision-making capabilities and facilitates proactive strategies to navigate the dynamic landscape of the analyzed data
Future Work	Further research could explore the integration of exogenous variables to enhance forecasting accuracy and consider alternative modeling techniques like deep learning architectures for capturing complex temporal dependencies. Additionally, investigating methods for uncertainty

T

	quantification and sensitivity analysis could provide valuable insights for more robust decision-making
	This project lays the groundwork for more accurate and proactive forecasting in various domains, facilitating informed decision-making and strategic planning in the face of dynamic temporal data.

Reviewer 1: Sri Checka Apparao	**Reviewer 2: Dr. D. Jaya Kumari.**

Trainer Name: Er. M. Rajesh kannan. ME.,

TIME SERIES ANALYSIS USING ELECTRIC PRODUCTION DATASET

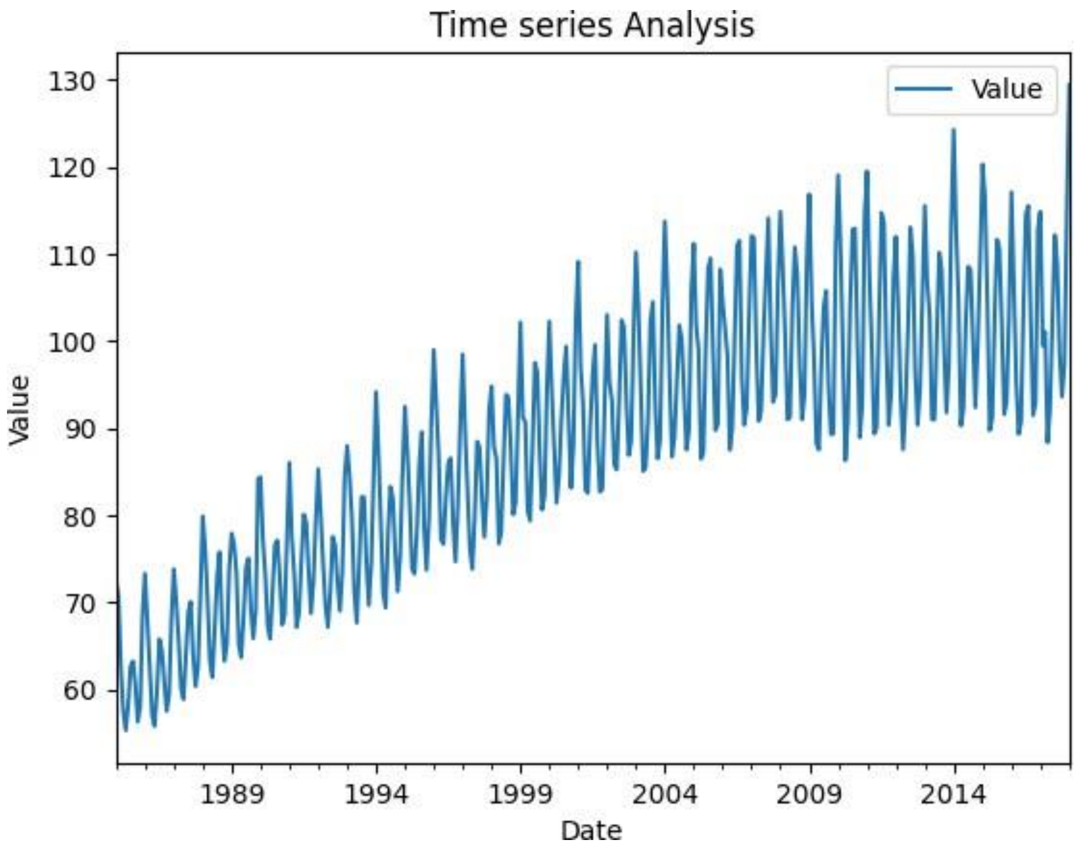
[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[4]:
```
#assuming data.csv contains time series data with a date column
df=pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.
↪csv',parse_dates=['DATE'],index_col='DATE')
```

[5]:
```
#Display first few rows
print(df.head())
# summary statstics
print(df.describe())
```

```
              Value
DATE
1985-01-01  72.5052
1985-02-01  70.6720
1985-03-01  62.4502
1985-04-01  57.4714
1985-05-01  55.3151
            Value
count  397.000000
mean    88.847218
std     15.387834
min     55.315100
25%     77.105200
50%     89.779500
75%    100.524400
max    129.404800
```

[6]:
```
#plot the time series data
df.plot()
plt.xlabel('Date')
plt.ylabel('Value')
plt.title("Time series Analysis")
plt.show()
```



[7]:
```
#select data for specific date range
subset=df['2014-01-01':'2014-12-31']
print(subset)
#select data for a specific year
subset=df['2014']
# select data for a specific month
subset=df['2014-01']
```

```
                Value
DATE
2014-01-01   124.2549
2014-02-01   112.8811
2014-03-01   104.7631
2014-04-01    90.2867
2014-05-01    92.1340
2014-06-01   101.8780
2014-07-01   108.5497
2014-08-01   108.1940
2014-09-01   100.4172
2014-10-01    92.3837
```

```
2014-11-01   99.7033
2014-12-01  109.3477
```

```
<ipython-input-7-10be6794ff0d>:5: FutureWarning: Indexing a DataFrame with a
datetimelike index using a single string to slice the rows, like
`frame[string]`, is deprecated and will be removed in a future version. Use
`frame.loc[string]` instead.
  subset=df['2014']
<ipython-input-7-10be6794ff0d>:7: FutureWarning: Indexing a DataFrame with a
datetimelike index using a single string to slice the rows, like
`frame[string]`, is deprecated and will be removed in a future version. Use
`frame.loc[string]` instead.
  subset=df['2014-01']
```

[8]:

```
# resample to monthly frequency and calculate the mean
monthly_mean=df.resample('M').mean()
print(monthly_mean)
```

```
               Value
DATE
1985-01-31   72.5052
1985-02-28   70.6720
1985-03-31   62.4502
1985-04-30   57.4714
1985-05-31   55.3151
…                …
2017-09-30   98.6154
2017-10-31   93.6137
2017-11-30   97.3359
2017-12-31  114.7212
2018-01-31  129.4048

[397 rows x 1 columns]
```

[9]:

```
# compute a 30-day rolling mean
rolling_mean=df.rolling(window=30).mean()
print(rolling_mean)
```

```
                 Value
DATE
1985-01-01         NaN
1985-02-01         NaN
1985-03-01         NaN
1985-04-01         NaN
1985-05-01         NaN
…                  …
2017-09-01  101.545190
2017-10-01  101.674110
```

```
2017-11-01  101.882207
2017-12-01  102.284597
2018-01-01  102.876910

[397 rows x 1 columns]
```

[10]:
```python
# shift the data by 1 day forward
shifted_forward=df.shift(1)
# shift data by 1 day backward
shifted_backward=df.shift(-1)
```

[11]:
```python
import pandas as pd

# Assuming df is your DataFrame containing datetime values

# Localize to UTC
df = df.tz_localize('UTC')
print(df)

# Convert to US/Eastern time zone
df = df.tz_convert('US/Eastern')
print(df)
```

```
                              Value
DATE
1985-01-01 00:00:00+00:00   72.5052
1985-02-01 00:00:00+00:00   70.6720
1985-03-01 00:00:00+00:00   62.4502
1985-04-01 00:00:00+00:00   57.4714
1985-05-01 00:00:00+00:00   55.3151
…                             …
2017-09-01 00:00:00+00:00   98.6154
2017-10-01 00:00:00+00:00   93.6137
2017-11-01 00:00:00+00:00   97.3359
2017-12-01 00:00:00+00:00  114.7212
2018-01-01 00:00:00+00:00  129.4048

[397 rows x 1 columns]
                              Value
DATE
1984-12-31 19:00:00-05:00   72.5052
1985-01-31 19:00:00-05:00   70.6720
1985-02-28 19:00:00-05:00   62.4502
1985-03-31 19:00:00-05:00   57.4714
1985-04-30 20:00:00-04:00   55.3151
…                             …
2017-08-31 20:00:00-04:00   98.6154
2017-09-30 20:00:00-04:00   93.6137
```

```
2017-10-31 20:00:00-04:00   97.3359
2017-11-30 19:00:00-05:00  114.7212
2017-12-31 19:00:00-05:00  129.4048

[397 rows x 1 columns]
```

[ ]:

```
import pandas as pd
import numpy as np
#generate a large volume of random data(eg. 1 millon data points)
volume_data =
pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.csv',
delimiter=',')
print("volume of data:",len(volume_data))
```

```
volume of data: 397
```

```
import time
#simulate streaming data every second for 10 seconds
velocity_data =
pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.csv',
sep =',')
for i in range(10):
  velocity_data=np.random.rand( )#generate random data
  print("velocity data point:",velocity_data)
  time.sleep(1)#wait for 1 second to simulate real-time data stream
```

```
velocity data point: 0.053739800040630836
velocity data point: 0.7600865628566403
velocity data point: 0.8334026740488973
velocity data point: 0.8611903492030597
velocity data point: 0.941339960538555
velocity data point: 0.7153484797213798
velocity data point: 0.0289426348532833
velocity data point: 0.045976217239983797
velocity data point: 0.03523363656495271
velocity data point: 0.3962593428101757
```

```
import pandas as pd
import random
#generate some random data with noise
veracity_data =
pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.csv',
sep =',')
veracity_data=[random.choice([1,2,3,None])for _ in range(20)]
print("veracity data:",veracity_data)
```

```
veracity data: [3, None, 2, 3, 3, 1, None, None, None, 1, 3, 3, 1, 1,
None, 2, 1, None, None, None]
```

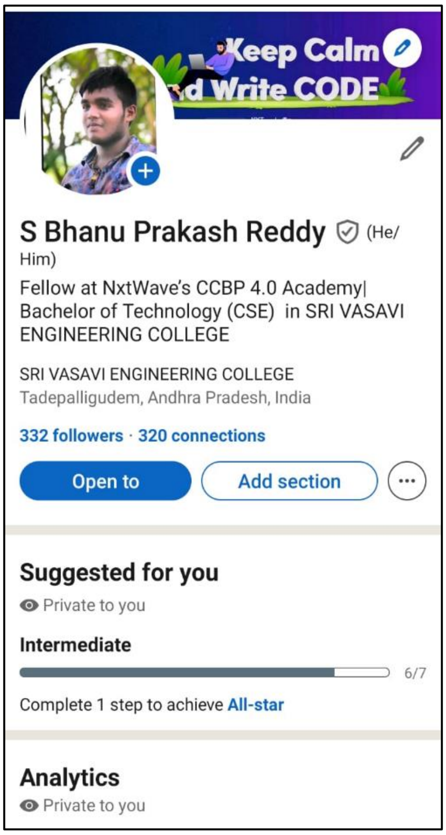
```
structured_data =
pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.csv',
sep =',')
#structured data
print(structured_data)
#semi-structured data(JSON)
semistructured_data =
pd.read_csv('/content/drive/MyDrive/dataset/Electric_Production.csv',
```

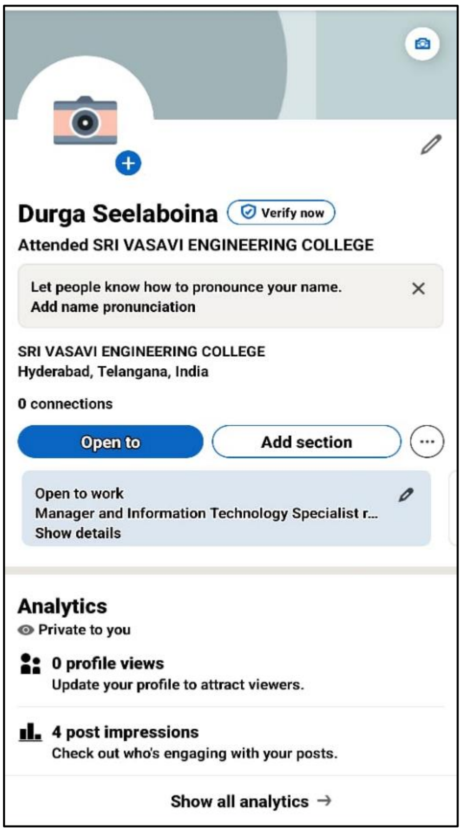
```
sep =',')
print(semistructured_data)
#unstructured data(text)
unstructured_data="This is a sample text document.It can contain any
information in free form"
```

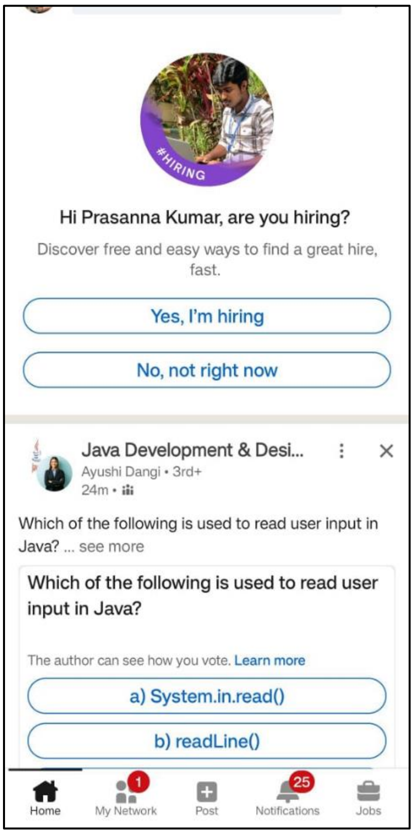
```
           DATE     Value
0    01-01-1985   72.5052
1    02-01-1985   70.6720
2    03-01-1985   62.4502
3    04-01-1985   57.4714
4    05-01-1985   55.3151
..          ...       ...
392  09-01-2017   98.6154
393  10-01-2017   93.6137
394  11-01-2017   97.3359
395  12-01-2017  114.7212
396  01-01-2018  129.4048

[397 rows x 2 columns]
           DATE     Value
0    01-01-1985   72.5052
1    02-01-1985   70.6720
2    03-01-1985   62.4502
3    04-01-1985   57.4714
4    05-01-1985   55.3151
..          ...       ...
392  09-01-2017   98.6154
393  10-01-2017   93.6137
394  11-01-2017   97.3359
395  12-01-2017  114.7212
396  01-01-2018  129.4048

[397 rows x 2 columns]
```

Social Media Handles

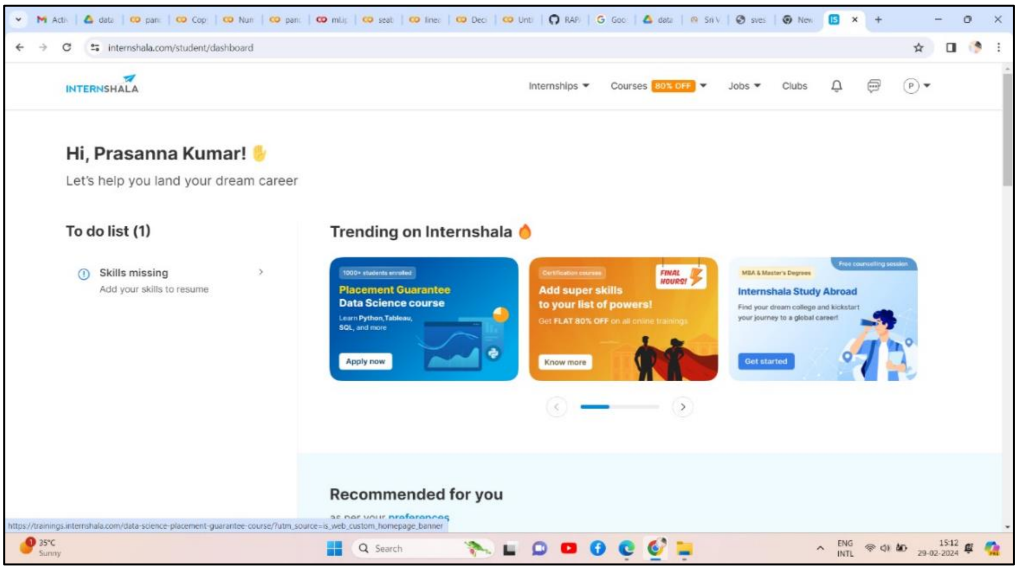
LinkedIn

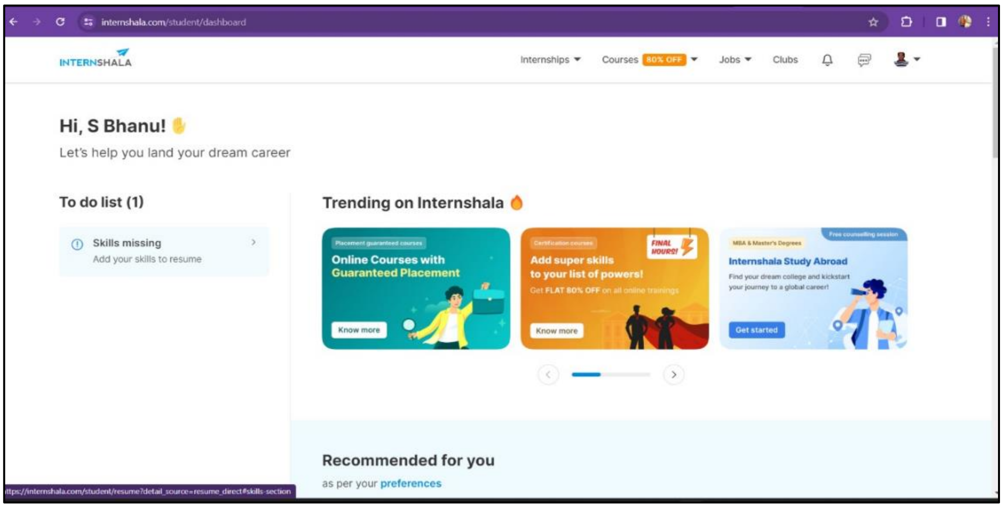


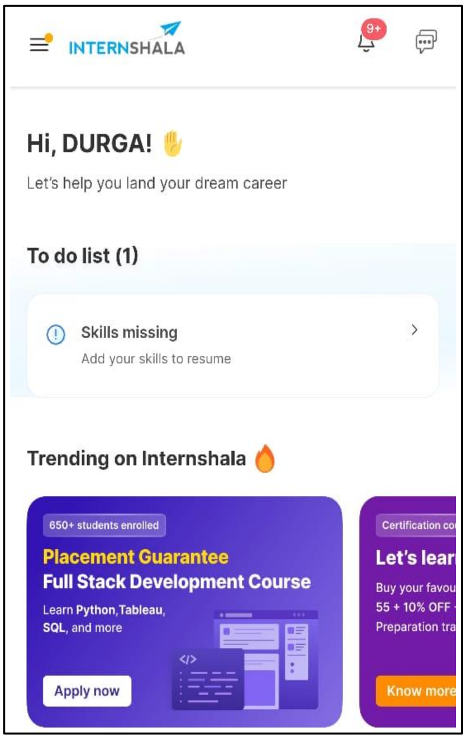




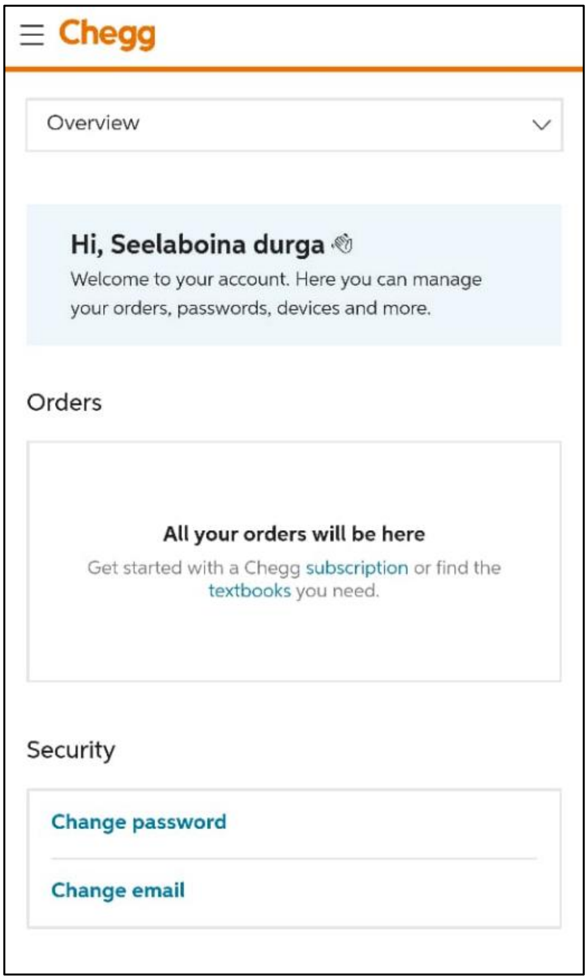
INTERNSHALA

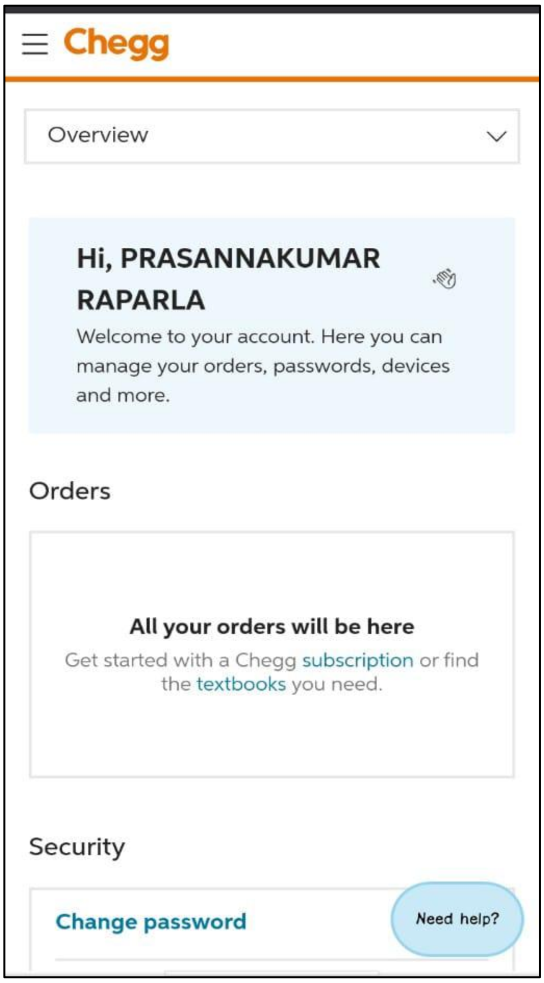


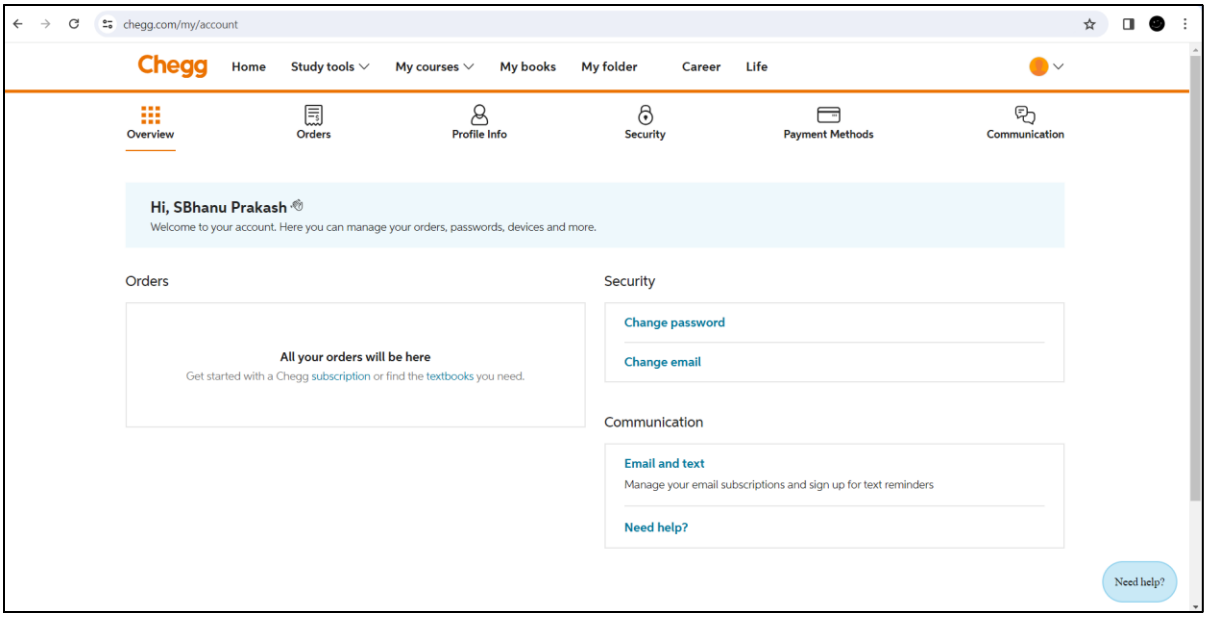




Chegg


Chegg
Overview
Hi, Seelaboina durga
Welcome to your account. Here you can manage your orders, passwords, devices and more.
Orders
All your orders will be here
Get started with a Chegg subscription or find the textbooks you need.
Security
Change password
Change email


Chegg
Overview
Hi, PRASANNAKUMAR RAPARLA
Welcome to your account. Here you can manage your orders, passwords, devices and more.
Orders
All your orders will be here
Get started with a Chegg subscription or find the textbooks you need.
Security
Change password
Need help?


chegg.com/my/account
Chegg
Home
Study tools
My courses
My books
My folder
Career
Life
Overview
Orders
Profile Info
Security
Payment Methods
Communication
Hi, SBhanu Prakash
Welcome to your account. Here you can manage your orders, passwords, devices and more.
Orders
All your orders will be here
Get started with a Chegg subscription or find the textbooks you need.
Security
Change password
Change email
Communication
Email and text
Manage your email subscriptions and sign up for text reminders
Need help?
Need help?

22A81A05I4
R.Harshitha

TEAM-6

LinkedIn POSTS

CSE-C

22A81A05I7
R.Prasanna Kumar

22A81A05I9
S. Mahathi

22A81A05I8
S.Navya Sri

22A81A05J2
T.VijayRatnam

22A81A05J0
S.Bhanu Prakash

22A81A05J1
S.Durga

LINEAR -ROJECT

[70]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt import
sklearn
```

[71]:
```
df=pd.read_csv('/content/drive/MyDrive/dataset/lpro.csv') df
```

[71]:

	Experience_Years	Age
0	5	28
1	1	21
2	3	23
3	2	22
4	1	17
5	25	62
6	19	54
7	2	21
8	10	36
9	15	54
10	4	26
11	6	29
12	14	39
13	11	40
14	2	23
15	4	27
16	10	34
17	15	54
18	2	21
19	10	36
20	15	54
21	4	26
22	5	29
23	1	21
24	4	23
25	3	22
26	1	18
27	27	62

28	19	54
29	2	21
30	10	34
31	15	54
32	20	55
33	19	53
34	16	49

[72]:
```
df.head(5)
```

[72] :

	Experience_Years	Age
0	5	28
1	1	21
2	3	23
3	2	22
4	1	17

[73] :
```
df.isna().sum()
```

[73] :
```
Experience_Years    0
Age                 0
dtype: int64
```

[74] :
```
X = df.iloc[:,:-1].values y =
df.iloc[:, 1].values
```

[77]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣

    ↪random_state =0 )
```

[78]:
```
from sklearn.linear_model import LinearRegression regressor =
LinearRegression()
print(y_train) regressor.fit(X_train, y_train)
```

```
[23 29 62 22 49 55 18 34 36 40 62 54 23 54 23 21 39 54 21 17 21 26 36 54
 21 53 22 28]
```

[78] : LinearRegression()

[79] :
```
y_pred = regressor.predict(X_test)
```

[80] :
```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue',label='Regression␣

    ↪Line')
plt.title('Experience vs Age (Training set)')
plt.xlabel('Years of Experience')
```

```
plt.ylabel('Age') plt.legend()
plt.show()
```

[83]:
```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue',label='Regression␣

    ↪Line')
plt.title('Experience vs Age (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Age')
plt.legend()
plt.show()
```


Experience vs Age (Test set)
Regression Line
70
60
50
40
30
20
Age
0
5
10
15
20
25
Years of Experience

<u>WEBSCRAPPING PROJECT REPORT ON :</u>   <u>TOP 10 YOUTUBE CHANNELS</u>

<u>DESCRIPTION:</u>

Youtube channels plays a vital role in society for all information like

Education, gaming, entertainment etc

In this project there is information on top 10 youtube channels and their subscriber count

<u>WEBSCRAPPING PROJECT REPORT ON :</u> **<u>PYTHON LISTS TABLE</u>**

DESCRIPTION:

In this project there is information on some of the python list methods

<u>WEBSCRAPPING PROJECT REPORT ON :</u> <u>CENSUS OF INDIA IN 2011</u>

<u>DESCRIPTION:</u>

In this project there is information on list of states with population, Sex ratio and density area in 2011.

WEB-SCRAPPING PROJECT ON

INDIA.GOV.IN website :

ABOUT:

The MyGov.in website serves as a platform for citizens to engage with the Indian government, facilitating two-way communication. It hosts various initiatives, discussions, surveys, and campaigns aimed at fostering citizen participation in governance. Users can access government announcements, contribute ideas, provide feedback, and collaborate on national initiatives through this interactive portal.

GITHUB QR CODE:

WEBSCRAPPING PROJECT ON: TOP INDIAN MOVIES

DESCRIPTION:

Indian movies play a multifaceted role in society, blending entertainment with cultural preservation, social commentary, and economic growth, making them an integral part of both Indian and global culture.

In this project there is information on top Indian films with their gross and budget by using webscrapping

LIST OF ALL PRIME MINISTER IN INDIA:

DESCRIPTION:

The Prime Minister is the head of the government and leader of the Council of Ministers. He is the chief of the executive branch of the Union Government. Since India's Independence in 1947, individuals of great calibre, with the passion to work for the country, have become Prime Ministers.

WEB-SCRAPPING PROJECT ON

WIKEPEDIA website :

ABOUT:

South Africa has nine provinces, each with its own capital. Here is a list of the South African provincial capitals:

Eastern Cape - Bhisho
Free State - Bloemfontein
Gauteng - Johannesburg
KwaZulu-Natal - Pietermaritzburg
Limpopo - Polokwane
Mpumalanga - Nelspruit (Mbombela)
North West - Mahikeng
Northern Cape - Kimberley
Western Cape - Cape Town
These capitals serve as the administrative centers for their respective provinces.

GITHUB QR CODE:

WEBSCRAPPING PROJECT ON:TOP10 CHEAPEST CURRENCIES:

DESCRIPTION:

We all have read and heard so much about the strongest currencies in the world. The British Pound Sterling, US Dollar, Swiss Franc, and Euro among others are the most well-known currencies in the world. The countries that issue these currencies are very stable, and this follows for their currencies as well. Not many of us know of the cheapest currency in the world and the countries that issue it.

WEB-SCRAPPING PROJECT ON

ALLTHATGROWS website :

ABOUT:

AllThatGrows is a comprehensive online platform offering a diverse range of high-quality seeds, bulbs, and gardening essentials. Their website provides an extensive collection of organic and heirloom seeds, along with expert gardening tips and resources. Inside, users can find detailed product information, growing guides, and a vibrant community forum to support their gardening endeavours.

GITHUB QR CODE: