LAB No. 1

Introduction to VS Code and Google Colab for Data Analysis Using Python

LAB Description

In this lab, students will learn how to work with Python programming environments using Visual Studio Code (VS Code) and Google Colab. VS Code is a lightweight and powerful source-code editor that runs on a local system and supports Python development through extensions. It allows users to write, run, and debug Python programs efficiently on their own computer.

Google Colab is a cloud-based Python notebook environment provided by Google that runs in a web browser. It does not require any local installation and provides free access to computing resources. Colab is especially useful for data analysis and visualization, as it supports interactive notebooks, built-in libraries, and easy file uploads.

Using either VS Code or Google Colab, students will create a dataset in Python, upload or load the data, and perform basic data analysis operations. The lab focuses on understanding how datasets are handled, how simple statistics are calculated, and how graphical representations help in interpreting data.

Lab Objective

The objectives of this lab are:

- To understand the basic functioning of VS Code and Google Colab
- To learn how to create and upload a dataset using Python
- To perform basic statistical analysis (such as mean, median, and count)
- To visualize data using simple graphs (line charts, bar charts, or histograms)
- To develop foundational skills in data analysis and visualization using Python

How to use python in VS code?

1. Create environment named 'venv'

```
python -m venv
```

2. Activate environment

```
venv\Scripts\activate
```

3. Select Environment in VS CodeAfter creating the environment:

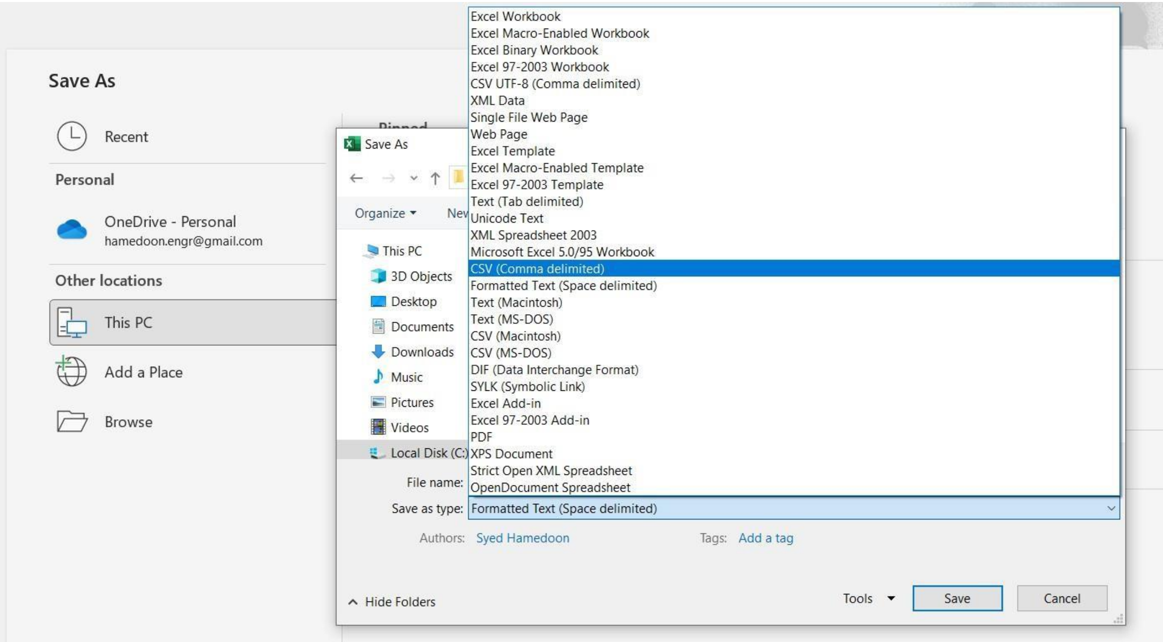
1) Open your project folder in VS Code.
2) Press Ctrl + Shift + P → search for Python:
3) Select Interpreter.
4) Choose your newly created environment (venv or myenv).

4. Install Machine Learning and Deep Learning Libraries

- pip install –upgrade pip
- pip install pandas
- pip install matplotlib
- pip install seaborn
- pip install scikit-learn

- pip install scipy
- pip install numpy
- pip install xgboost
- pip install lightgbm
- pip install catboost
- pip install tensorflow
- pip install keras
- pip install torch

Task 1:
Save Excel file as .csv (Comma delimited)

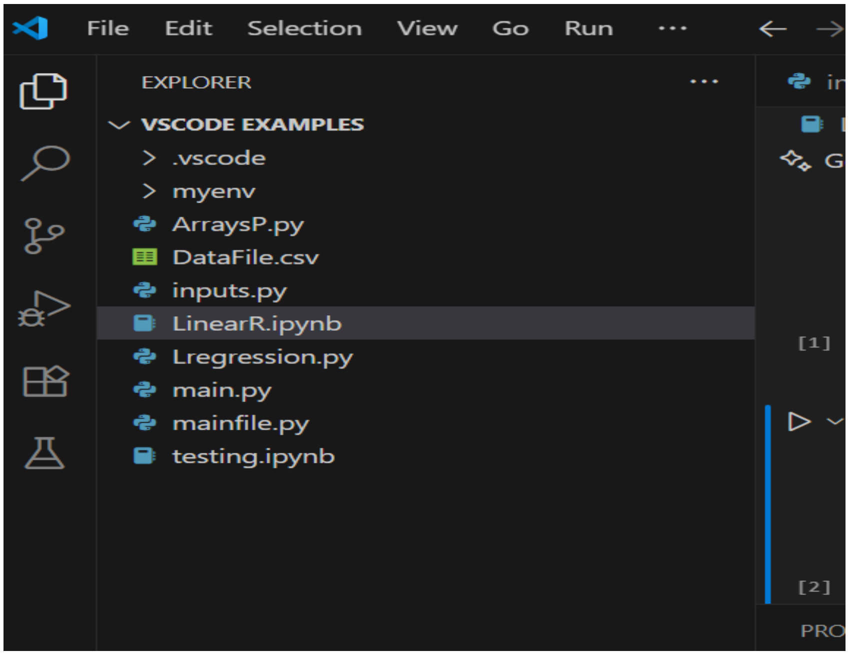


Save as type: Formatted Text (Space delimited)

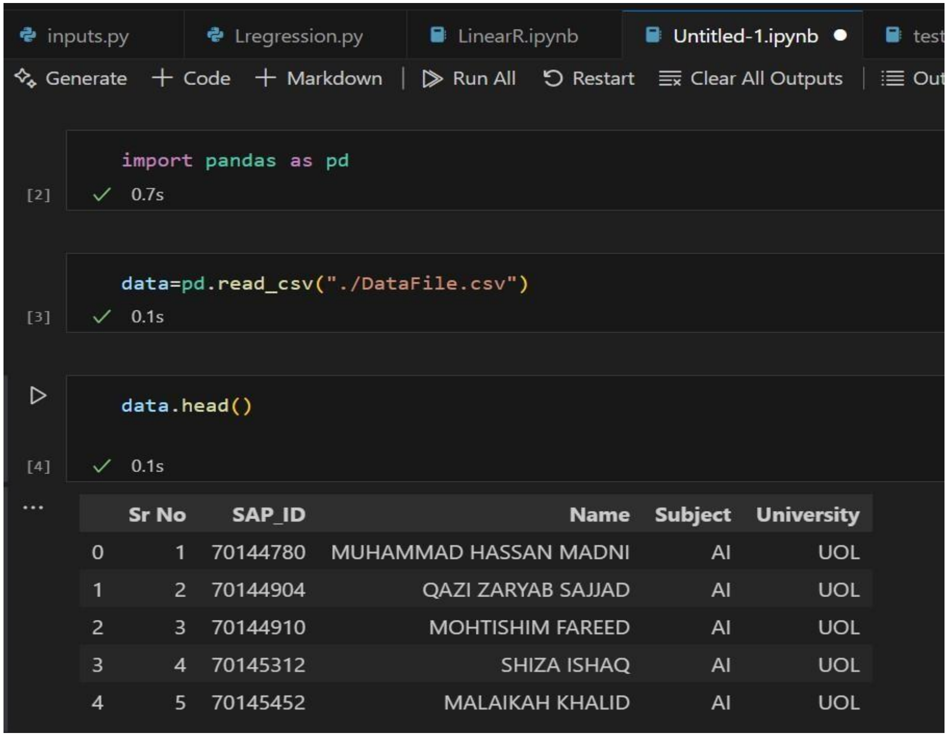
Authors: Syed Hamedoon

Tags: Add a tag

Task 2:
Save .csv file in directory and folder where we created a python environment



Here we can see our file successfully



```
[2]  import pandas as pd
     ✓ 0.7s
```

```
[3]  data=pd.read_csv("./DataFile.csv")
     ✓ 0.1s
```

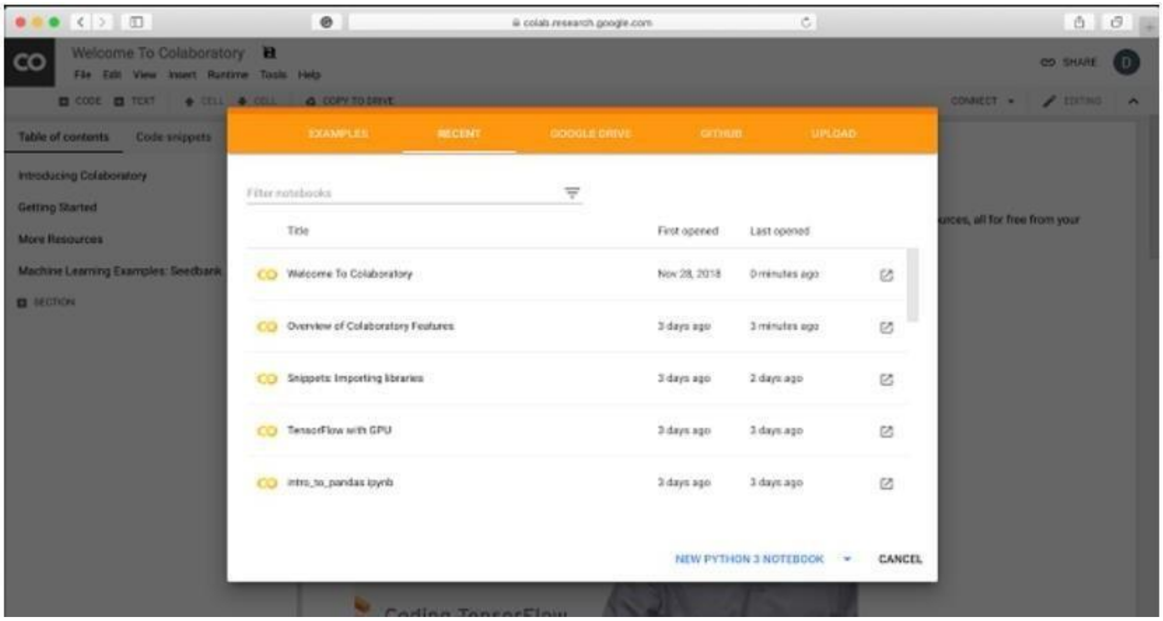
```
[4]  data.head()
     ✓ 0.1s
```

	Sr No	SAP_ID	Name	Subject	University
0	1	70144780	MUHAMMAD HASSAN MADNI	AI	UOL
1	2	70144904	QAZI ZARYAB SAJJAD	AI	UOL
2	3	70144910	MOHTISHIM FAREED	AI	UOL
3	4	70145312	SHIZA ISHAQ	AI	UOL
4	5	70145452	MALAIKAH KHALID	AI	UOL

Introduction of Google Colab

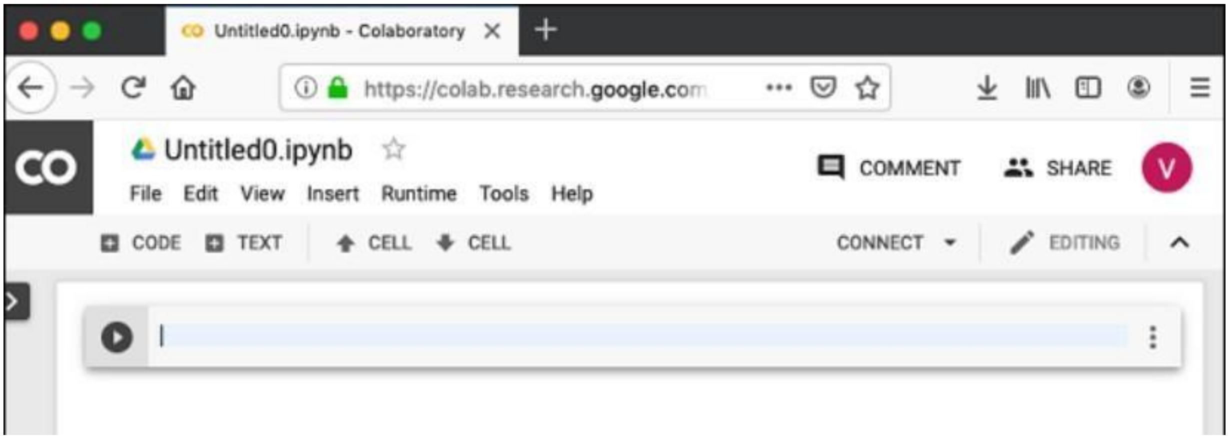
- Colab is a free notebook environment that runs entirely in the cloud. It lets you and your team members edit documents, the way you work with Google Docs.

- Colab supports many popular machine learning libraries which can be easily loaded in your notebook.
- Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science.
- Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook
- **Step 1** – Open the following URL in your browser – https://colab.research.google.com Your browser would display the following screen (assuming that you are logged into your Google Drive) –



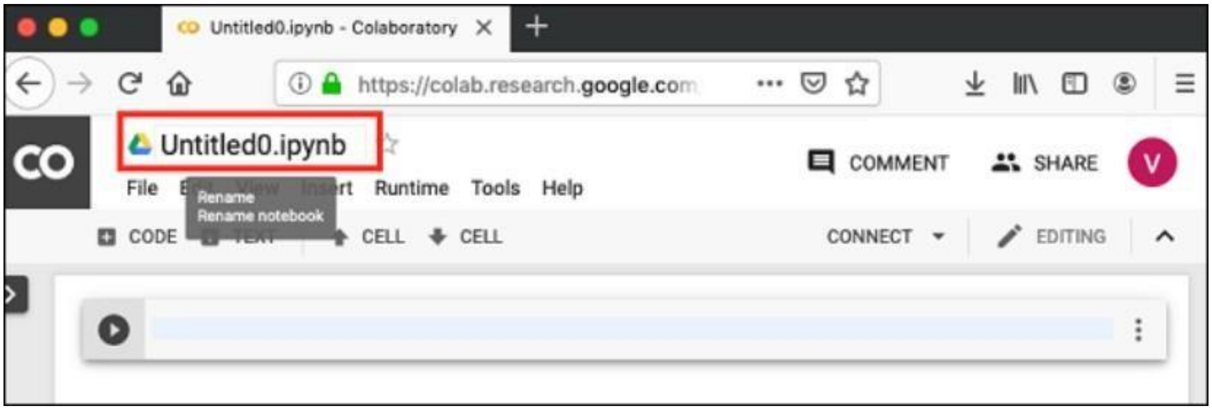
Title	First opened	Last opened
Welcome To Colaboratory	Nov 28, 2018	0 minutes ago
Overview of Colaboratory Features	3 days ago	3 minutes ago
Snippets: Importing libraries	3 days ago	2 days ago
TensorFlow with GPU	3 days ago	3 days ago
intro_to_pandas.ipynb	3 days ago	3 days ago

- **Step 2** – Click on the **NEW PYTHON 3 NOTEBOOK** link at the bottom of the screen. A new notebook would open up as shown in the screen below.

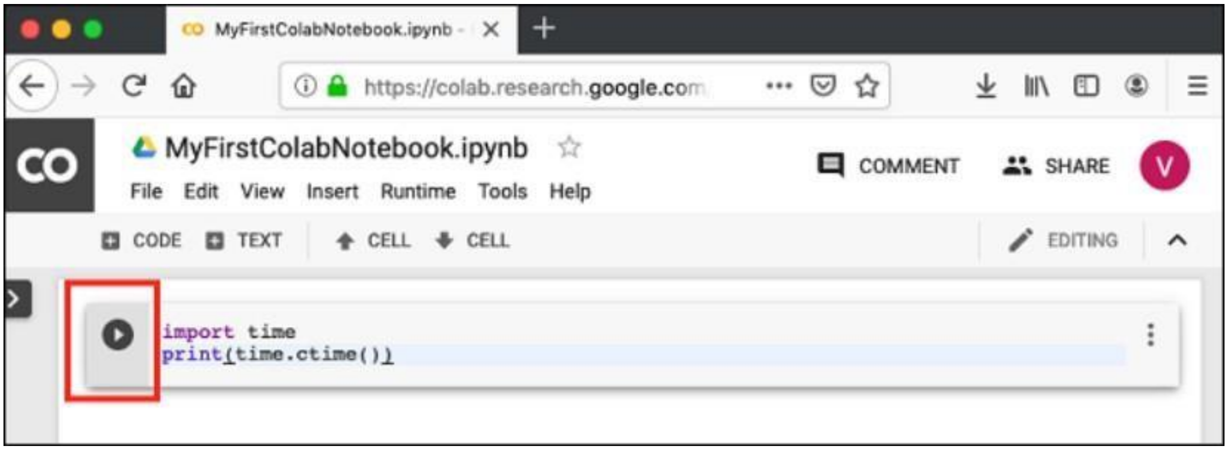

Untitled0.ipynb - Colaboratory
https://colab.research.google.com
CO
Untitled0.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
V
CODE
TEXT
CELL
CELL
CONNECT
EDITING

- **Setting Notebook Name**

  By default, the notebook uses the naming convention UntitledXX.ipynb. To rename the notebook, click on this name and type in the desired name in the edit box as shown here –

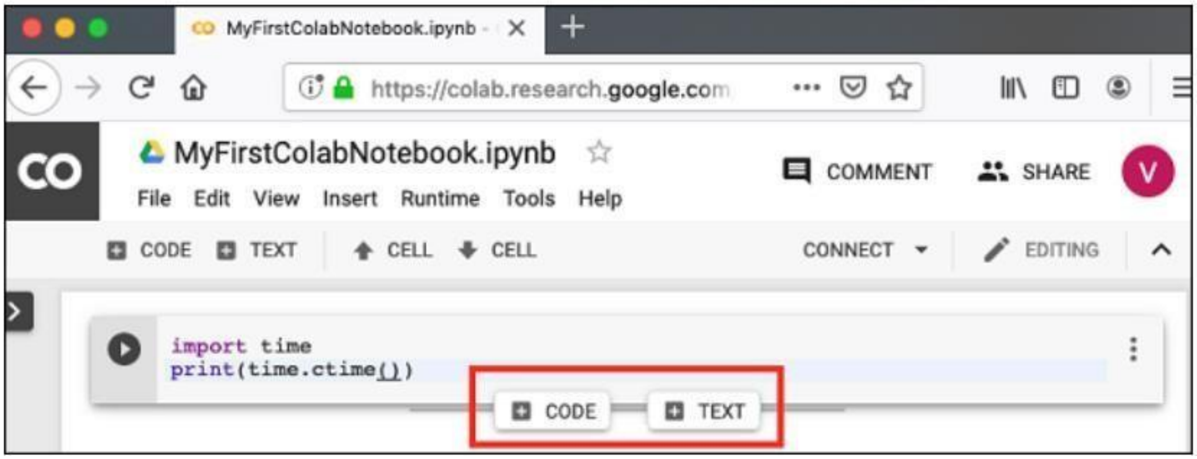

Untitled0.ipynb - Colaboratory
https://colab.research.google.com
CO
Untitled0.ipynb
File Edit View Insert Runtime Tools Help
Rename
Rename notebook
COMMENT
SHARE
V
CODE
TEXT
CELL
CELL
CONNECT
EDITING

- Entering Code


MyFirstColabNotebook.ipynb -
https://colab.research.google.com
CO
MyFirstColabNotebook.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
V
CODE
TEXT
CELL
CELL
EDITING

```
import time
print(time.ctime())
```

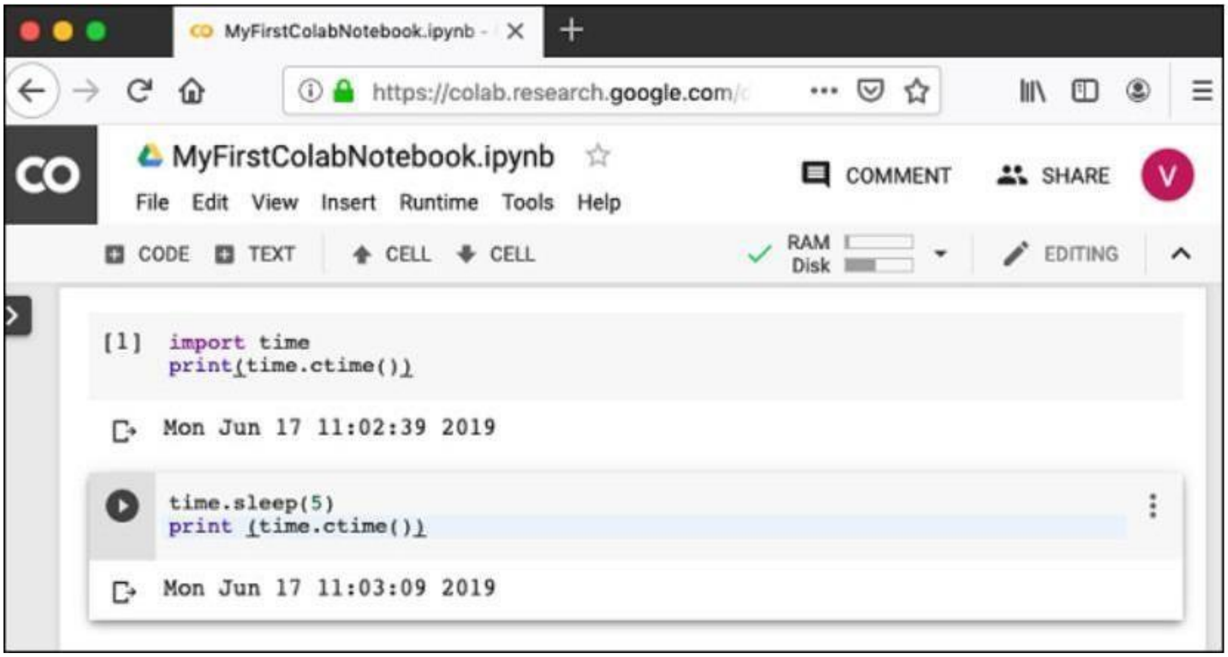
- Adding Code Cells
- To add more code to your notebook, select the following **menu** options –


MyFirstColabNotebook.ipynb -
https://colab.research.google.com
CO
MyFirstColabNotebook.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
V
CODE
TEXT
CELL
CELL
CONNECT
EDITING
CODE
TEXT

```
import time
print(time.ctime())
```

- **Run All**

  To run the entire code in your notebook without an interruption, execute the following menu options −


MyFirstColabNotebook.ipynb -
https://colab.research.google.com/
CO
MyFirstColabNotebook.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
V
CODE
TEXT
CELL
CELL
RAM
Disk
EDITING

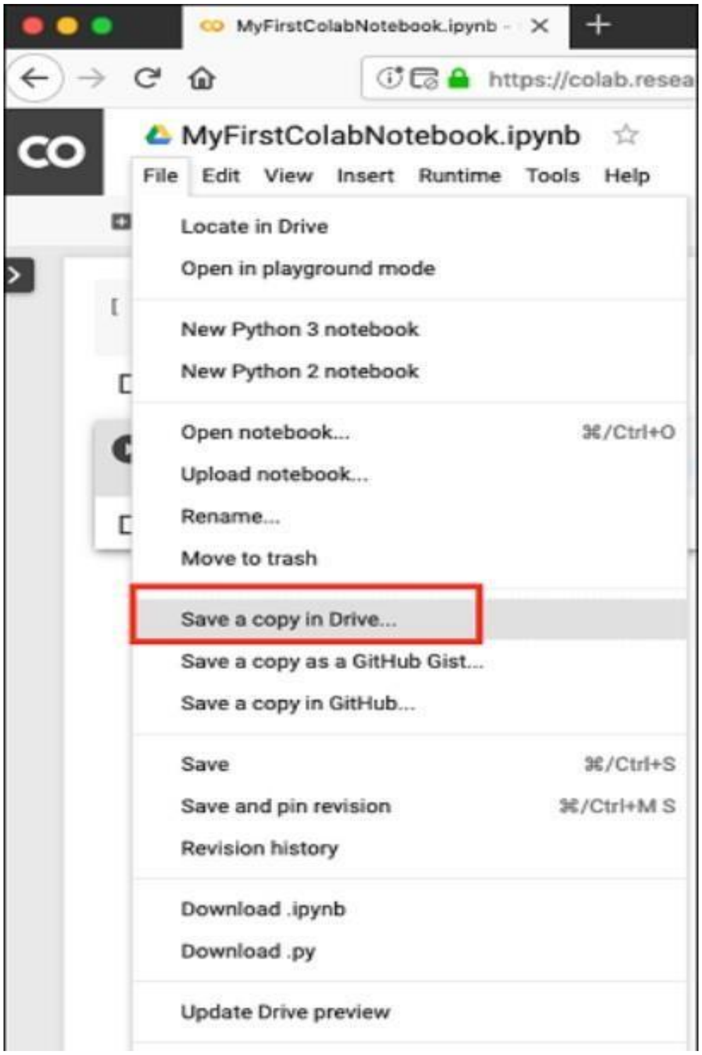
```
[1] import time
    print(time.ctime())

Mon Jun 17 11:02:39 2019
```

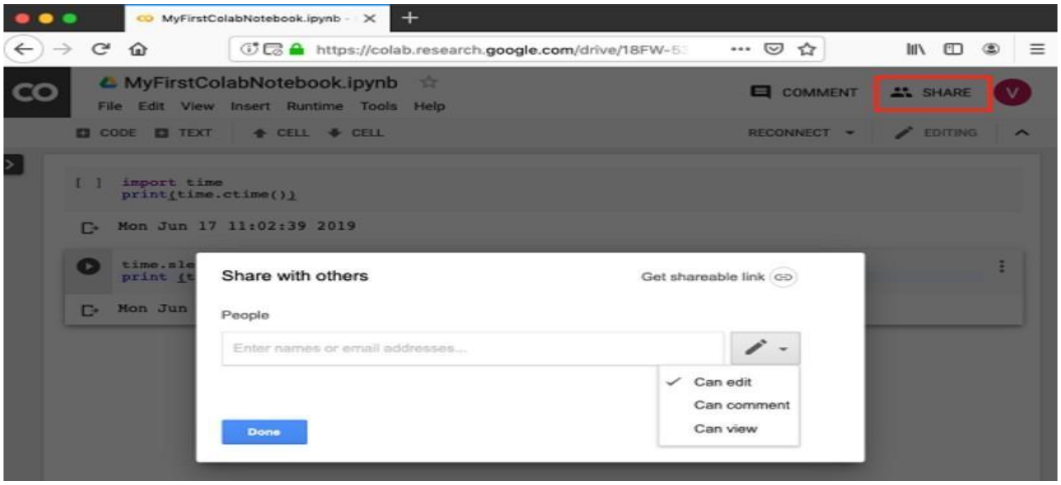
```
time.sleep(5)
print (time.ctime())

Mon Jun 17 11:03:09 2019
```

Saving to Google Drive



Google Colab - Sharing Notebook



```
[ ] import time
    print(time.ctime())
```

```
Mon Jun 17 11:02:39 2019
```

```
time.sle
print (t
```

```
Mon Jun
```

LAB Assignment No. 1

Lab Assignment – Dataset Creation s Analysis

Objective

To learn how to create and upload a dataset in Python, perform basic statistical analysis, and visualize data using graphs.

Tasks

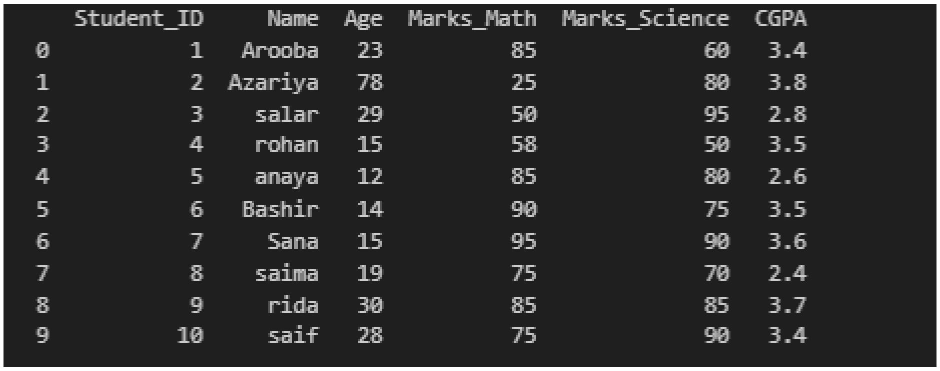
🔷 Q1: Create a Dataset Manually

- Create a dataset of at least **10 students** with the following columns:
  - Student_ID,
  - Name,
  - Age,
  - Marks_Math,
  - Marks_Science.
- Store the dataset in a **CSV file** named students.csv.

Code:

```
import pandas as pd
data = {
   'Student_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
   'Name': ['Arooba', 'Azariya', 'salar', 'rohan', 'anaya', 'Bashir', 'Sana', 'saima', 'rida', 'saif'],
   'Age': [23, 78, 29, 15, 12, 14, 15, 19, 30, 28],
   'Marks_Math': [85, 25, 50, 58, 85, 90, 95, 75, 85, 75],
   'Marks_Science': [60, 80, 95, 50, 80, 75, 90, 70, 85, 90],
   'CGPA': [3.4, 3.8, 2.8, 3.5, 2.6, 3.5, 3.6, 2.4, 3.7, 3.4]
}
df = pd.DataFrame(data)
print(df)
```

Output:



```
   Student_ID     Name  Age  Marks_Math  Marks_Science  CGPA
0           1   Arooba   23          85             60   3.4
1           2  Azariya   78          25             80   3.8
2           3    salar   29          50             95   2.8
3           4    rohan   15          58             50   3.5
4           5    anaya   12          85             80   2.6
5           6   Bashir   14          90             75   3.5
6           7     Sana   15          95             90   3.6
7           8    saima   19          75             70   2.4
8           9     rida   30          85             85   3.7
9          10     saif   28          75             90   3.4
```

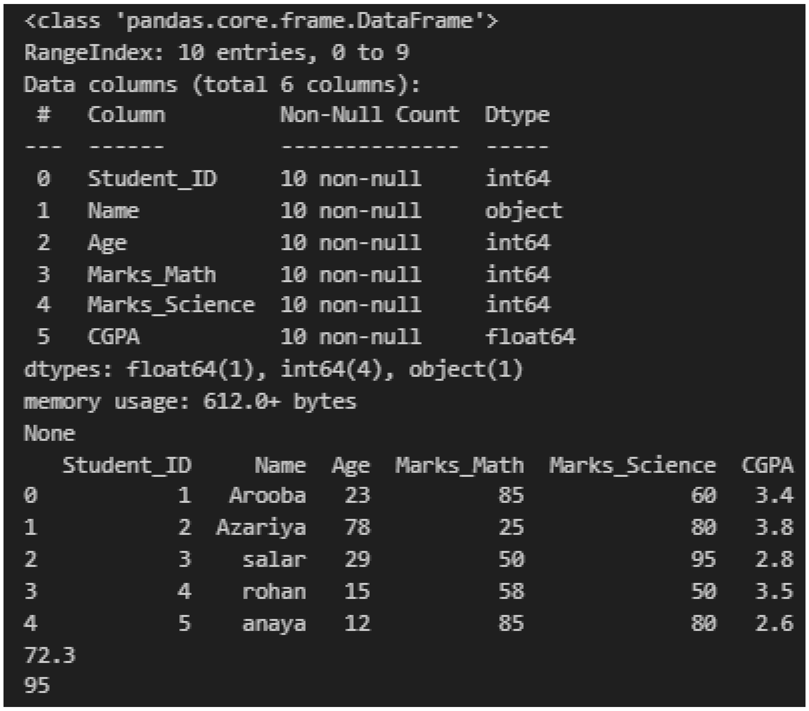
Q2: Upload Dataset in Python

- Use **Pandas** to load the dataset.

Code:

```
print(df.info())
print(df.head())
print(df['Marks_Math'].mean())
print(df['Marks_Science'].max())
```

Output:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Student_ID     10 non-null     int64  
 1   Name           10 non-null     object 
 2   Age            10 non-null     int64  
 3   Marks_Math     10 non-null     int64  
 4   Marks_Science  10 non-null     int64  
 5   CGPA           10 non-null     float64
dtypes: float64(1), int64(4), object(1)
memory usage: 612.0+ bytes
None
   Student_ID     Name  Age  Marks_Math  Marks_Science  CGPA
0           1   Arooba   23          85             60   3.4
1           2  Azariya   78          25             80   3.8
2           3    salar   29          50             95   2.8
3           4    rohan   15          58             50   3.5
4           5    anaya   12          85             80   2.6
72.3
95
```

Q3: Observe Dataset Information

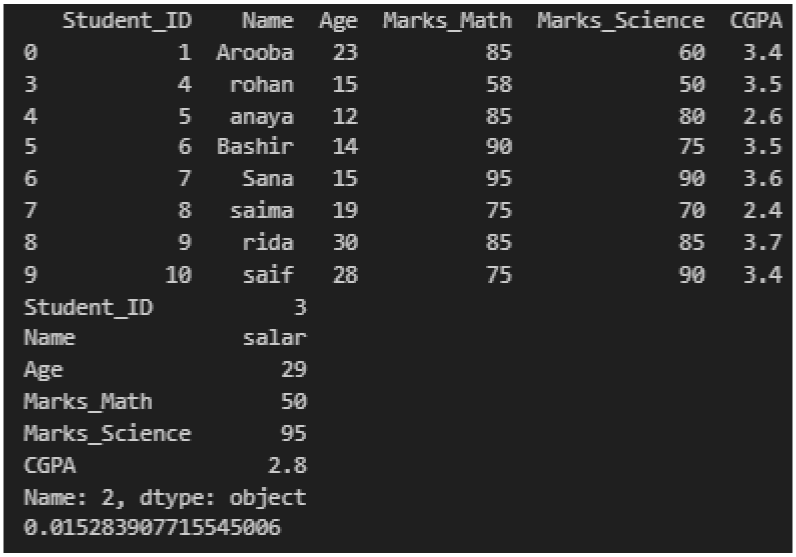
Run the following commands and explain the output:

1. data.info() → Dataset structure
2. data.describe() → Summary statistics (mean, std, min, max, etc.)
3. data['Marks_Math'].mean() → Mean of Math marks
4. data['Marks_Science'].max() → Maximum Science marks

Code:

```
print(df[df['Marks_Math'] > 50])
print(df.loc[df['Marks_Science'].idxmax()])
print(df['Marks_Math'].corr(df['Marks_Science']))
```

output:



```
   Student_ID    Name  Age  Marks_Math  Marks_Science  CGPA
0           1  Arooba   23          85             60   3.4
3           4   rohan   15          58             50   3.5
4           5   anaya   12          85             80   2.6
5           6  Bashir   14          90             75   3.5
6           7    Sana   15          95             90   3.6
7           8   saima   19          75             70   2.4
8           9    rida   30          85             85   3.7
9          10    saif   28          75             90   3.4
Student_ID           3
Name             salar
Age                 29
Marks_Math          50
Marks_Science       95
CGPA               2.8
Name: 2, dtype: object
0.015283907715545006
```

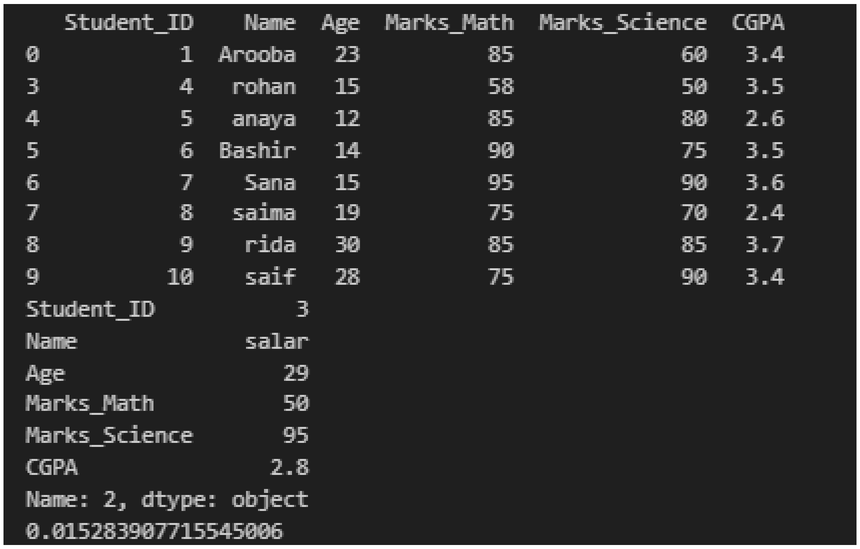
Q4: Perform Some Data Analysis

- Find how many students have Marks_Math > 50.
- Find the student with the **highest Science marks**.
- Calculate the **correlation** between Marks_Math and Marks_Science.

Code:

```
print(df[df['Marks_Math'] > 50])
print(df.loc[df['Marks_Science'].idxmax()])
print(df['Marks_Math'].corr(df['Marks_Science']))
```

Output:



```
   Student_ID    Name  Age  Marks_Math  Marks_Science  CGPA
0           1  Arooba   23          85             60   3.4
3           4   rohan   15          58             50   3.5
4           5   anaya   12          85             80   2.6
5           6  Bashir   14          90             75   3.5
6           7    Sana   15          95             90   3.6
7           8   saima   19          75             70   2.4
8           9    rida   30          85             85   3.7
9          10    saif   28          75             90   3.4
Student_ID           3
Name             salar
Age                 29
Marks_Math          50
Marks_Science       95
CGPA               2.8
Name: 2, dtype: object
0.015283907715545006
```

Q5: Data Visualization

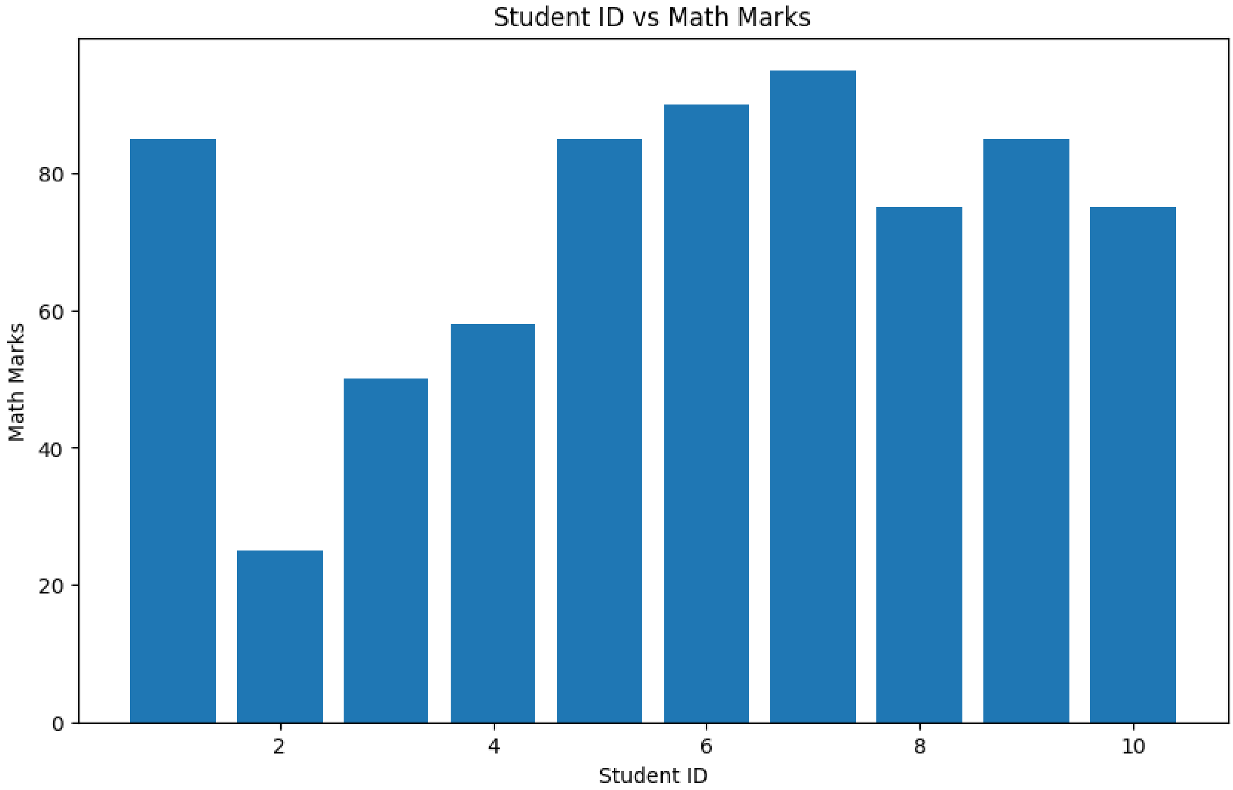
Use **Matplotlib/Seaborn** to create graphs:

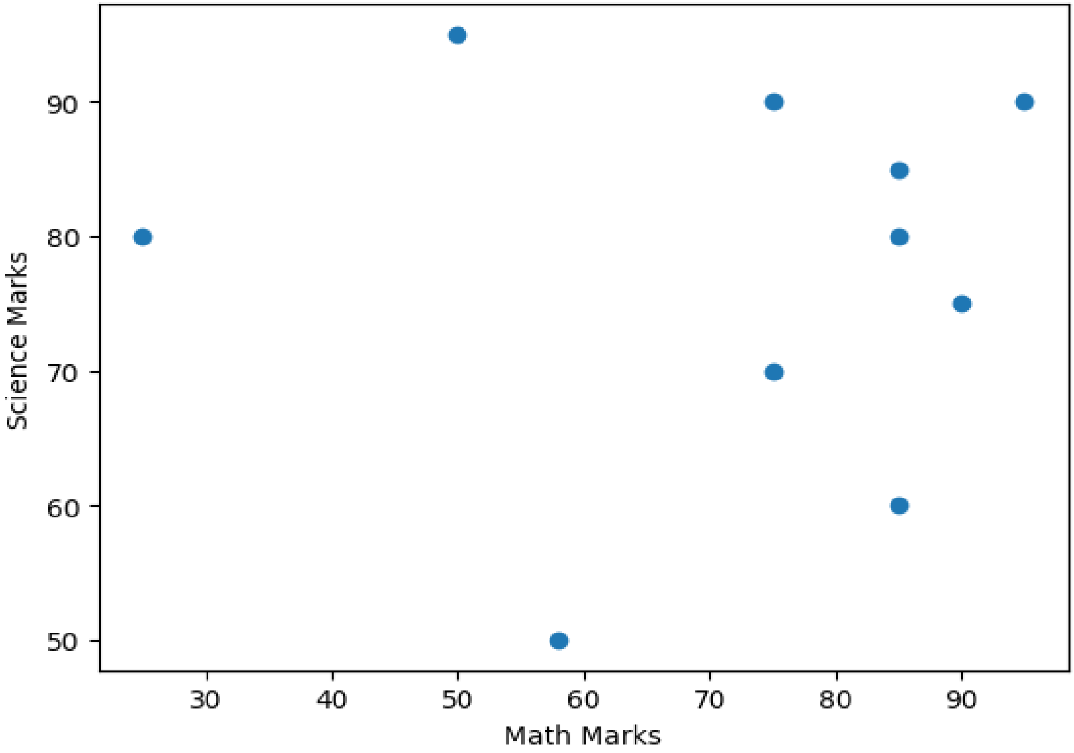
1. A bar chart of Student_ID vs Marks_Math.
2. A histogram of Age.
3. A scatter plot of Marks_Math vs Marks_Science.

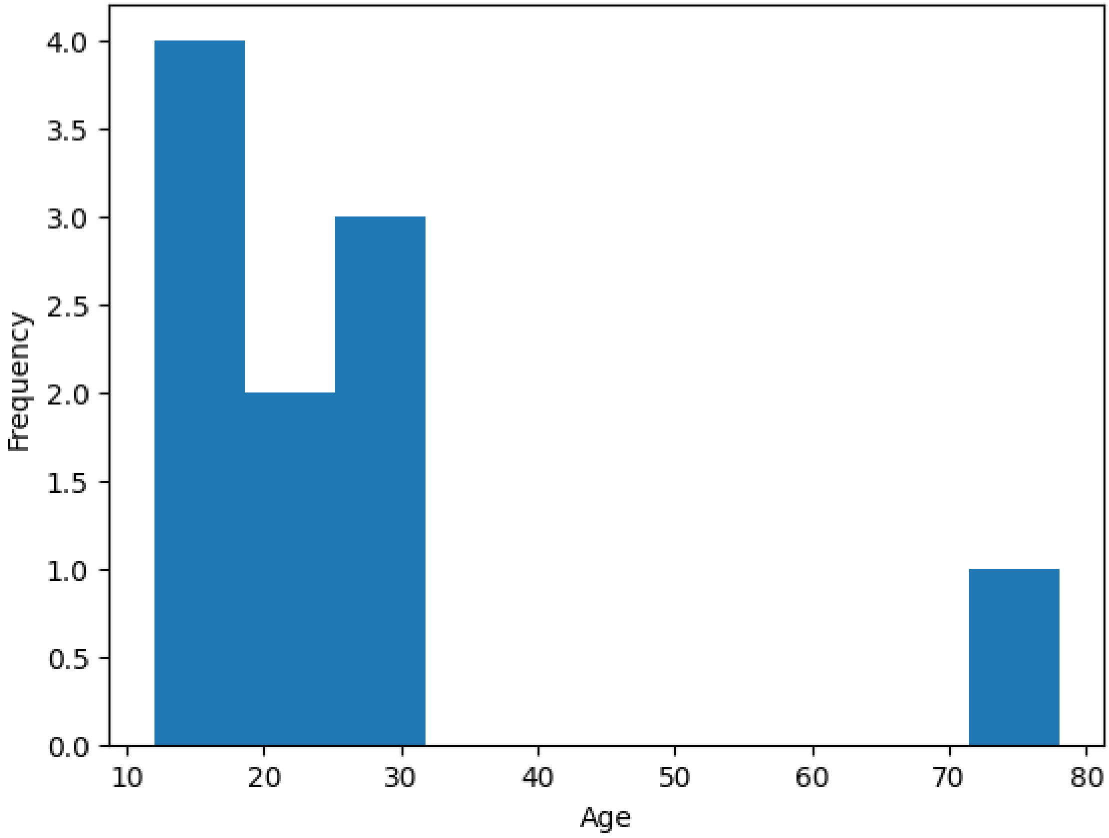
Code:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.bar(df['Student_ID'], df['Marks_Math'])
plt.xlabel('Student ID')
plt.ylabel('Math Marks')
plt.title('Student ID vs Math Marks')
plt.show()
import matplotlib.pyplot as plt
plt.scatter(df['Marks_Math'], df['Marks_Science'])
plt.xlabel('Math Marks')
plt.ylabel('Science Marks')
plt.show()
plt.hist(df['Age'])
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Output:


Student ID vs Math Marks
Math Marks
0
20
40
60
80
Student ID
2
4
6
8
10


Science Marks
50
60
70
80
90
Math Marks
30
40
50
60
70
80
90



LAB Assessment

Student Name		**LAB Rubrics**	CLO3 , P5, PLO5
		Total Marks	10
Registration No		**Obtained Marks**	
		Teacher Name	Dr. Syed M Hamedoon
Date		**Signature**	

Laboratory Work Assessment Rubrics

Sr. No.	Performance Indicator	Excellent (5)	Good (4)	Average (3)	Fair (2)	Poor (1)
1	**Theoretical knowledge 10%**	Student knows all the related concepts about the theoretical background of the experiment and rephrase those concepts in written and oral assessments	Student knows most of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows few of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows very little about the related concepts about the theoretical background of the experiment and poorly rephrase those concepts in written and oral assessments	Student has poor understanding of the related concepts about the theoretical background of the experiment and unable to rephrase those concepts in written and oral assessments
2	**Application Functionality 10%**	Application runs smoothly and operation of the application runs efficiently	Application compiles with no warnings. Robust operation of the application, with good recovery.	Application compiles with few or no warnings. Consideration given to unusual conditions with reasonable	Application compiles and runs without crashing. Some attempt at detecting and correcting errors.	Application does not compile or compiles but crashes. Confusing. Little or no error detection or correction.
3	**Specifications 10%**	The program works very efficiently and meets all of the required specifications.	The program works and meets some of the specifications.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	The program produces correct results but does not display them correctly.	The program is producing incorrect results.
4	**Level of understanding of the learned skill 10%**	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner with few errors	Provide partially correct and logical answers based upon minimum technical content to the questions asked by examiner	Provide very few and illogical answers to the questions asked by examiner.	Provide no answer to the questions asked by examiner.
5	**Readability and Reusability 10%**	The code is exceptionally well organized and very easy to follow and reused	The code is fairly easy to read. The code could be reused as a whole or each class could be reused.	Most of the code could be reused in other programs.	Some parts of the code require change before they could be reused in other programs.	The code is poorly organized and very difficult to read and not organized for reusability.

6	**AI System Design 10%**	Well-designed AI models. Code is highly maintainable	Good designed AI models and Little code duplications	Some attempt to make AI models. Code can be maintained with significant effort	Little attempt to design AI models and less understanding of code	Very poor attempt to design AI models and its code
7	**Responsiveness to Questions/ Accuracy 10%**	1. Responds well, quick and very accurate all the time. 2. Effectively uses eye contact, speaks clearly, effectively and confidently using suitable volume	1. Generally Responsive and accurate most of the times. 2. Maintains eye contact, speaks clearly with suitable volume and pace.	1. Generally Responsive and accurate few times. 2. Some eye contact, speaks clearly and unclearly in different portions.	1. Not much Responsive and accurate most of the times. 2. Uses eye contact ineffectively and fails to speak clearly and audibly	. 1. Non Responsive and inaccurate all the times. 2. No eye contact and unable to speak 3. Dresses inappropriately
8	**Efficiency 10%**	The code is extremely efficient without sacrificing readability and understanding	The code is fairly efficient without sacrificing readability and understanding	Some part of the code is efficient and other part of the code is not understandable and work properly	The code is brute force and unnecessarily long	The code is huge and appears to be patched together
9	**Delivery** **10%**	The program was delivered in time during lab.	The program was delivered in Lab before the end time.	The program was delivered within the due date.	The code was delivered within a day after the due date.	The code was delivered more than 2 days overdue.
10	**Awareness of Safety Guidelines 10%**	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is fully compliant to the guidelines	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines	Student has no knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines