HOMEWORK 1

Question 1

1.Perform 6 hyperlink level navigation using browser use AI on your academic use

SOLUTION

INPUT

Start from https://chanakyauniversity.edu.in (label as Step 0). Then perform exactly 6 hyperlink clicks on this site, following this path if available:

1. School of Engineering
2. Computer Science & Artificial Intelligence
3. Programme Structure (or closest curriculum link)
4. Admissions
5. Fee Structure
6. Hostel Facilities (or Hostels)

After each click, output exactly: Step N: <Page Title>, URL<full URL>:

If any label isn't visible, choose the closest equivalent and continue. Do not skip steps. Do not summarize.

OUTPUT

Step 0: Chanakya University | Admission 2025
URL: https://chanakyauniversity.edu.in

Step 1: Seeking a path to pursue your degree?
URL: https://chanakyauniversity.edu.in/school-of-engineering

Step 2: Seeking a path to pursue your degree ?
URL: https://chanakyauniversity.edu.in/programs/btech-computer-science-and-artificial-intelligence/

Step 3: Seeking a path to pursue your degree ?
URL: https://chanakyauniversity.edu.in/programs/btech-computer-science-and-artificial-intelligence/

Step 4: Chanakya University | Admission 2025
URL: https://chanakyauniversity.edu.in/admissions/

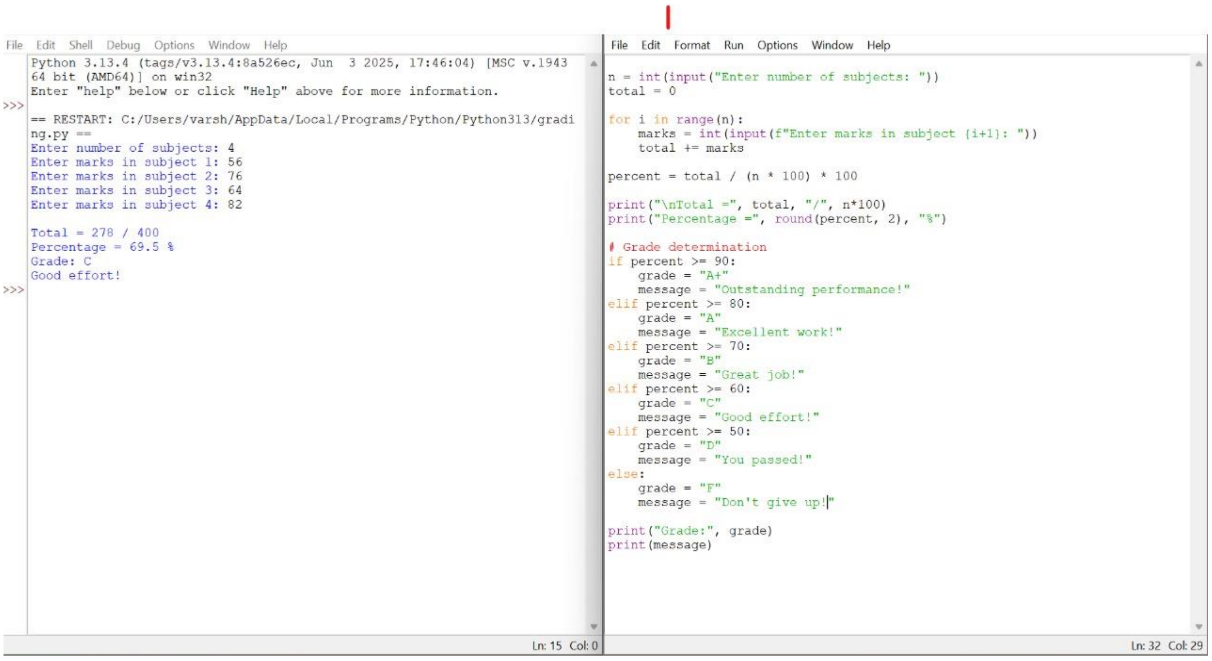
Step 5: Chanakya University | Admission 2025
URL: https://chanakyauniversity.edu.in/fee-structure/

Step 6: Chanakya University | Admission 2025
URL: https://chanakyauniversity.edu.in/residential-life/

Question 2

2.Build your own AI assistant for grading a student based on the subjects and a marks entered

SOLUTION


File Edit Shell Debug Options Window Help
File Edit Format Run Options Window Help
Ln: 15 Col: 0
Ln: 32 Col: 29

```
Python 3.13.4 (tags/v3.13.4:8a526ec, Jun  3 2025, 17:46:04) [MSC v.1943
64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
== RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python313/gradi
ng.py ==
Enter number of subjects: 4
Enter marks in subject 1: 56
Enter marks in subject 2: 76
Enter marks in subject 3: 64
Enter marks in subject 4: 82

Total = 278 / 400
Percentage = 69.5 %
Grade: C
Good effort!
>>>
```

```python
n = int(input("Enter number of subjects: "))
total = 0

for i in range(n):
    marks = int(input(f"Enter marks in subject {i+1}: "))
    total += marks

percent = total / (n * 100) * 100

print("\nTotal =", total, "/", n*100)
print("Percentage =", round(percent, 2), "%")

# Grade determination
if percent >= 90:
    grade = "A+"
    message = "Outstanding performance!"
elif percent >= 80:
    grade = "A"
    message = "Excellent work!"
elif percent >= 70:
    grade = "B"
    message = "Great job!"
elif percent >= 60:
    grade = "C"
    message = "Good effort!"
elif percent >= 50:
    grade = "D"
    message = "You passed!"
else:
    grade = "F"
    message = "Don't give up!"

print("Grade:", grade)
print(message)
```

INPUT

```python
n = int(input("Enter number of subjects: "))

total = 0

for i in range(n):

    marks = int(input(f"Enter marks in subject {i+1}: "))

    total += marks

percent = total / (n * 100) * 100

print("\nTotal =", total, "/", n*100)

print("Percentage =", round(percent, 2), "%")
```

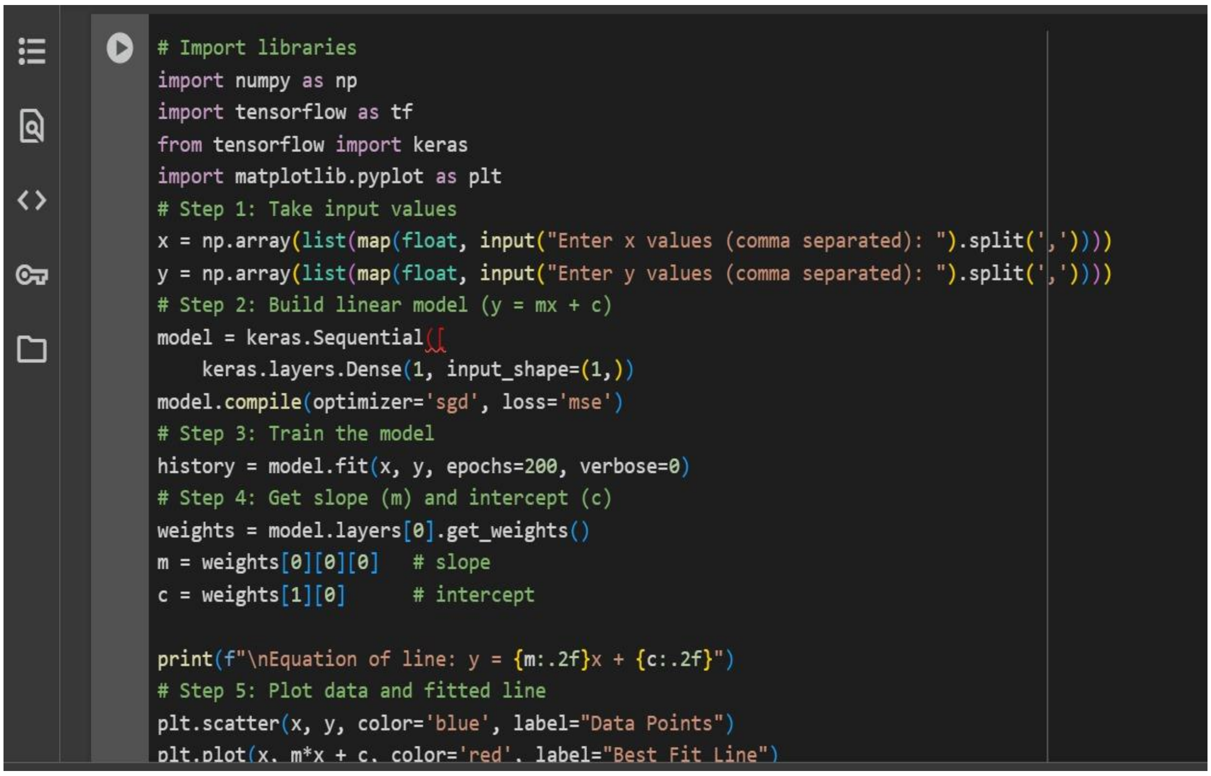
```
# Grade determination
if percent >= 90:
    grade = "A+"
    message = "Outstanding performance!"
elif percent >= 80:
    grade = "A"
    message = "Excellent work!"
elif percent >= 70:
    grade = "B"
    message = "Great job!"
elif percent >= 60:
    grade = "C"
    message = "Good effort!"
elif percent >= 50:
    grade = "D"
    message = "You passed!"
else:
    grade = "F"
    message = "Don't give up!"
print("Grade:", grade)
print(message)
```

Question 3

3.Create a linear model using TensorFlow for dynamic line equation y=mx+c (your learning from Programming with Python to be applied as well)

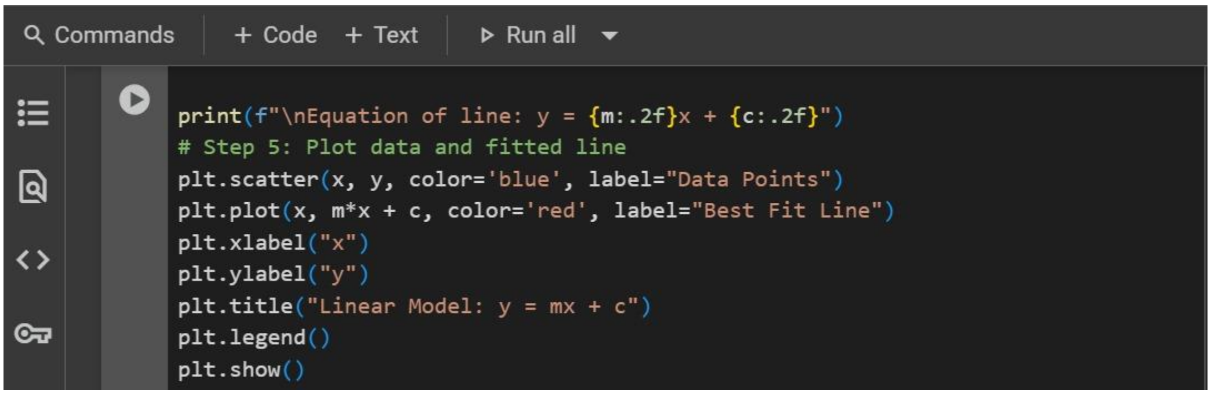
SOLUTION

INPUT



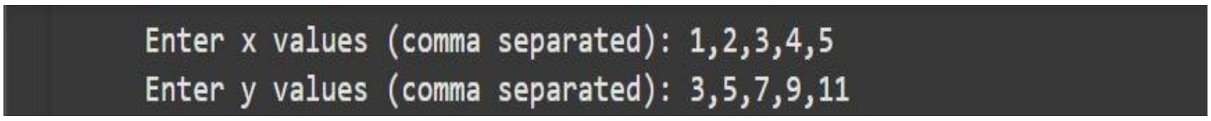
```python
# Import libraries
import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
# Step 1: Take input values
x = np.array(list(map(float, input("Enter x values (comma separated): ").split(','))))
y = np.array(list(map(float, input("Enter y values (comma separated): ").split(','))))
# Step 2: Build linear model (y = mx + c)
model = keras.Sequential([
    keras.layers.Dense(1, input_shape=(1,))
model.compile(optimizer='sgd', loss='mse')
# Step 3: Train the model
history = model.fit(x, y, epochs=200, verbose=0)
# Step 4: Get slope (m) and intercept (c)
weights = model.layers[0].get_weights()
m = weights[0][0][0]   # slope
c = weights[1][0]      # intercept

print(f"\nEquation of line: y = {m:.2f}x + {c:.2f}")
# Step 5: Plot data and fitted line
plt.scatter(x, y, color='blue', label="Data Points")
plt.plot(x, m*x + c, color='red', label="Best Fit Line")
```

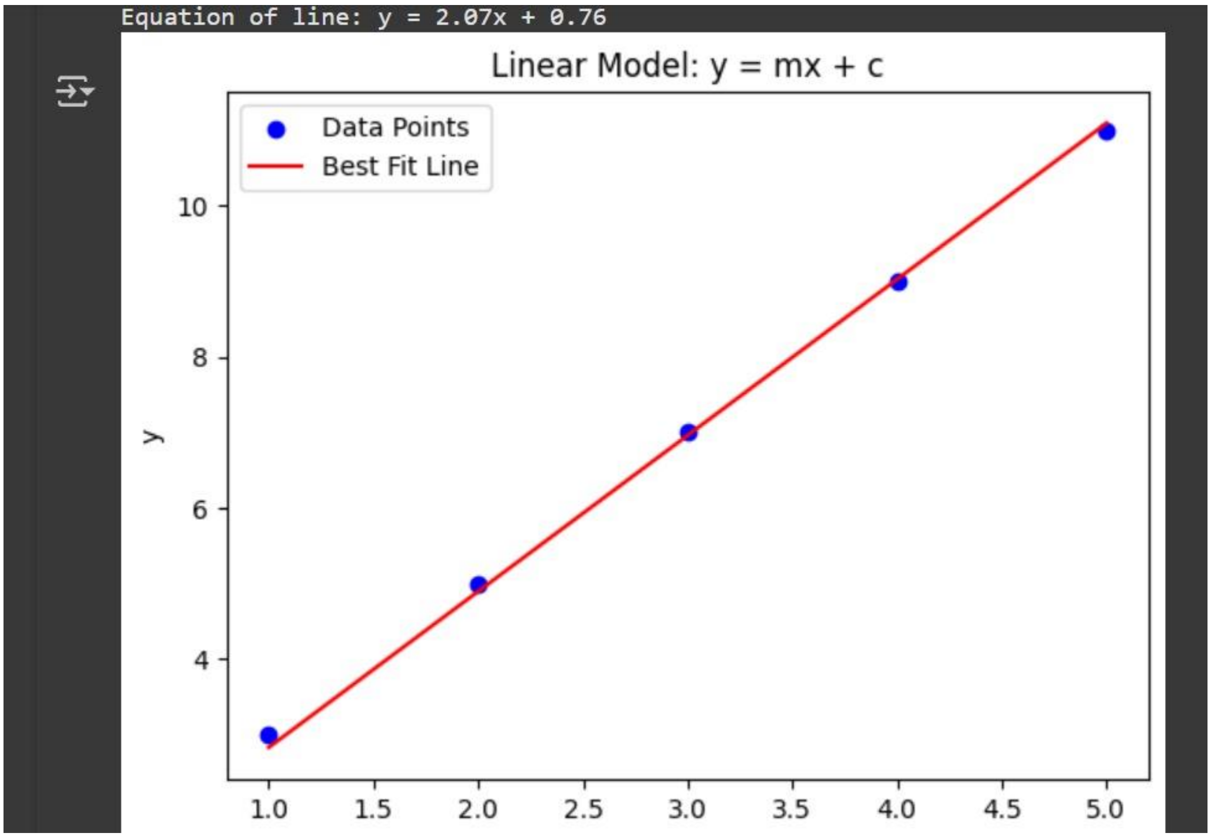


```python
print(f"\nEquation of line: y = {m:.2f}x + {c:.2f}")
# Step 5: Plot data and fitted line
plt.scatter(x, y, color='blue', label="Data Points")
plt.plot(x, m*x + c, color='red', label="Best Fit Line")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Linear Model: y = mx + c")
plt.legend()
plt.show()
```

OUTPUT



```
Enter x values (comma separated): 1,2,3,4,5
Enter y values (comma separated): 3,5,7,9,11
```



```
Equation of line: y = 2.07x + 0.76
```

CODE

```
# Import libraries
import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
# Step 1: Take input values
x = np.array(list(map(float, input("Enter x values (comma separated): ").split(','))))
y = np.array(list(map(float, input("Enter y values (comma separated): ").split(','))))
# Step 2: Build linear model (y = mx + c)
model = keras.Sequential([
    keras.layers.Dense(1, input_shape=(1,))
])
```

```
model.compile(optimizer='sgd', loss='mse')
# Step 3: Train the model
history = model.fit(x, y, epochs=200, verbose=0)
# Step 4: Get slope (m) and intercept (c)
weights = model.layers[0].get_weights()
m = weights[0][0][0]
c = weights[1][0]
print(f"\nEquation of line: y = {m:.2f}x + {c:.2f}")
# Step 5: Plot data and fitted line
plt.scatter(x, y, color='blue', label="Data Points")
plt.plot(x, m*x + c, color='red', label="Best Fit Line")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Linear Model: y = mx + c")
plt.legend()
plt.show()
```