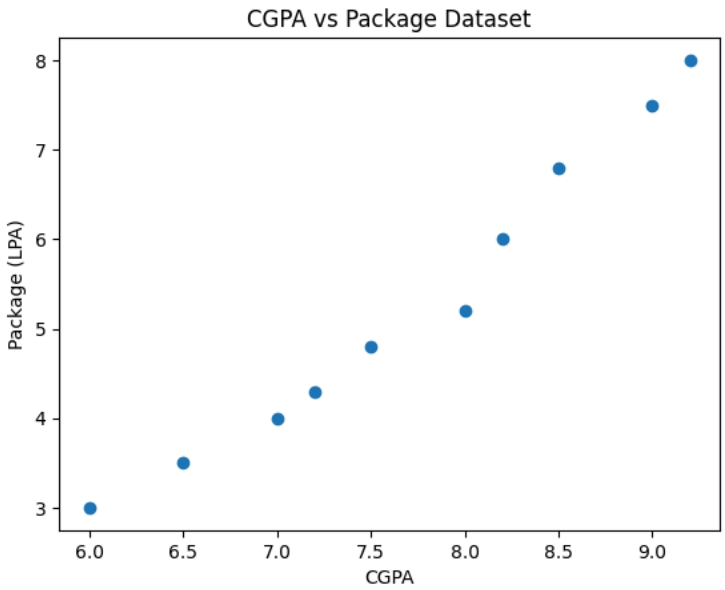
```
# Step 1: Import required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

```
# Step 2: Create dataset (10 students)
cgpa = np.array([6.0, 6.5, 7.0, 7.2, 7.5, 8.0, 8.2, 8.5, 9.0, 9.2]).reshape(-1, 1)
package = np.array([3.0, 3.5, 4.0, 4.3, 4.8, 5.2, 6.0, 6.8, 7.5, 8.0])
```

```
# Step 3: Scatter plot
plt.scatter(cgpa, package)
plt.xlabel("CGPA")
plt.ylabel("Package (LPA)")
plt.title("CGPA vs Package Dataset")
plt.show()
```




```
# Step 4: Create and train Linear Regression model
model = LinearRegression()
model.fit(cgpa, package)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Step 5: Take new CGPA input from user
new_cgpa = float(input("Enter CGPA to predict package: "))

# Step 6: Predict package
predicted_package = model.predict([[new_cgpa]])

# Step 7: Show result
print(f"Predicted Package for CGPA {new_cgpa} is: {predicted_package[0]:.2f}
LPA")
```

```
Enter CGPA to predict package: 8.3
Predicted Package for CGPA 8.3 is: 6.25 LPA
```

```
# Step 8: Display coefficient and intercept
print("Coefficient (slope):", model.coef_[0])
print("Intercept:", model.intercept_)
```

```
Coefficient (slope): 1.5962708146375515
Intercept: -6.997247980855523
```

```
# Step 9: Predict values for regression line
# Predict package values for all CGPA values to draw regression line
```

```
predicted_line = model.predict(cgpa)

# Step 10: Plot dataset and regression line
# Scatter plot shows actual data points
# Red line shows the best-fit regression line
plt.scatter(cgpa, package)
plt.plot(cgpa, predicted_line, color='red')
plt.xlabel("CGPA")
plt.ylabel("Package (LPA)")
plt.title("Linear Regression Model: CGPA vs Package")
plt.show()
```