In [1]:
```
#Import relevant functions and libraries

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
import seaborn as sns
```

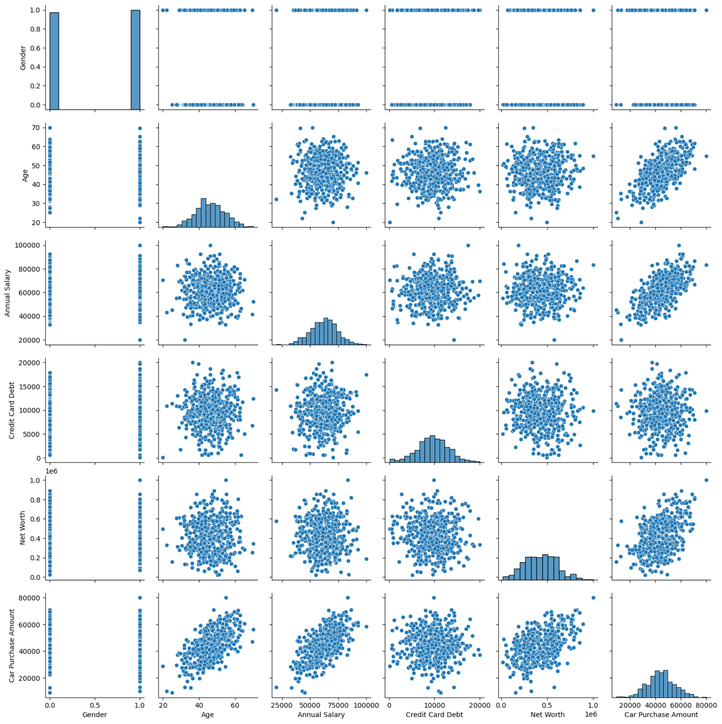
In [2]:
```
# Load the dataset
# This particular csv file is 'ISO-8859-1 encoded.

df = pd.read_csv('Car_Purchasing_Data.csv', encoding = 'ISO-8859-1')
```

In [3]:
```
# Plotting Correlation Charts using seaborn for having an overview of the data
sns.pairplot(df)
```

```
/Users/sachinpillai/anaconda3/lib/python3.11/site-packages/seaborn/axisgrid.py:118: UserWarning: The figure l
ayout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[3]:
```
<seaborn.axisgrid.PairGrid at 0x12c7afa50>
```



In [4]:
```
# Names, Customer E-mail and Country does not have a
# role to play in the model and is not taken in the training sets as variables


X = df[['Gender', 'Age', 'Annual Salary', 'Credit Card Debt', 'Net Worth']]
y = df['Car Purchase Amount']
```

In [5]:
```
X
```

Out[5]:

	Gender	Age	Annual Salary	Credit Card Debt	Net Worth
0	0	41.851720	62812.09301	11609.380910	238961.2505
1	0	40.870623	66646.89292	9572.957136	530973.9078
2	1	43.152897	53798.55112	11160.355060	638467.1773
3	1	58.271369	79370.03798	14426.164850	548599.0524
4	1	57.313749	59729.15130	5358.712177	560304.0671
...	...	...	...	...	...
495	0	41.462515	71942.40291	6995.902524	541670.1016
496	1	37.642000	56039.49793	12301.456790	360419.0988
497	1	53.943497	68888.77805	10611.606860	764531.3203
498	1	59.160509	49811.99062	14013.034510	337826.6382
499	1	46.731152	61370.67766	9391.341628	462946.4924

500 rows × 5 columns

In [6]:
```
X.shape
```

Out[6]:
```
(500, 5)
```

In [7]:
```
y
```

Out[7]:
```
0      35321.45877
1      45115.52566
2      42925.70921
3      67422.36313
4      55915.46248
          ...
495    48901.44342
496    31491.41457
497    64147.28888
498    45442.15353
499    45107.22566
Name: Car Purchase Amount, Length: 500, dtype: float64
```

In [8]:
```
y.shape
```

Out[8]:
```
(500,)
```

In [9]:
```
# Split the dataset into training and testing sets in 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [10]:
```
# Initialising and training the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[10]:
```
LinearRegression()
```

In [11]:
```
# Evaluate the model on the test set(20%)
predictions = model.predict(X_test)
print("Model Evaluation:")
print("Mean Absolute Error (MAE):", mean_absolute_error(y_test, predictions))
print("Mean Squared Error (MSE):", mean_squared_error(y_test, predictions))
print("Root Mean Squared Error (RMSE):", np.sqrt(mean_squared_error(y_test, predictions)))
print("R^2 Score:", r2_score(y_test, predictions))
```

```
Model Evaluation:
Mean Absolute Error (MAE): 1.1535708940625045
Mean Squared Error (MSE): 2.0943696024579865
Root Mean Squared Error (RMSE): 1.4471936990112921
R^2 Score: 0.9999999806028682
```

An R^2 Score of 0.9 is a good result

In [15]:
```
def predict_car_purchase_amount(gender, age, annual_salary, credit_card_debt, net_worth):
    # Create a DataFrame for the input data with the appropriate column names
    input_data_df = pd.DataFrame(data=[[gender, age, annual_salary, credit_card_debt, net_worth]],
                                 columns=['Gender', 'Age', 'Annual Salary', 'Credit Card Debt', 'Net Worth'])

    # Use the model to predict the car purchase amount using the DataFrame
    prediction = model.predict(input_data_df)
    return prediction[0]

# Input the required values
gender = 1  # Assuming 1 represents Female and 0 represents Male
age = 45
annual_salary = 80000
credit_card_debt = 9000
net_worth = 500000

# Evaluating the predicted purchase amount for the above values
predicted_amount = predict_car_purchase_amount(gender, age, annual_salary, credit_card_debt, net_worth)
print(f"Predicted Car Purchase Amount: {predicted_amount}")
```

```
Predicted Car Purchase Amount: 55201.364643797475
```

In [ ]:
```

```

In [ ]:
```

```