In [ ]:

```
# Name: Meet Hiteshkumar Trivedi
# Student Id: N1520331
```

Lab 7- Logistic Regression

Import Libraries

In [32]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Get the Data

**Read in the Import Libraries.csv file*

In [45]:

```
data = pd.read_csv('E:\Programming\Humber college\Humber Sem 2\Data Analytics\Week-10\Socia
```

Check the head of data

In [46]:

```
data.head()
```

Out[46]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [ ]:

In [47]:

```
#Write the code to show the result(below)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   User ID          400 non-null    int64 
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64 
 3   EstimatedSalary  400 non-null    int64 
 4   Purchased        400 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

In [48]:

```
data.describe()
```

Out[48]:

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

In [49]:

```
#Write the code to show the result(below)
```

In [50]:

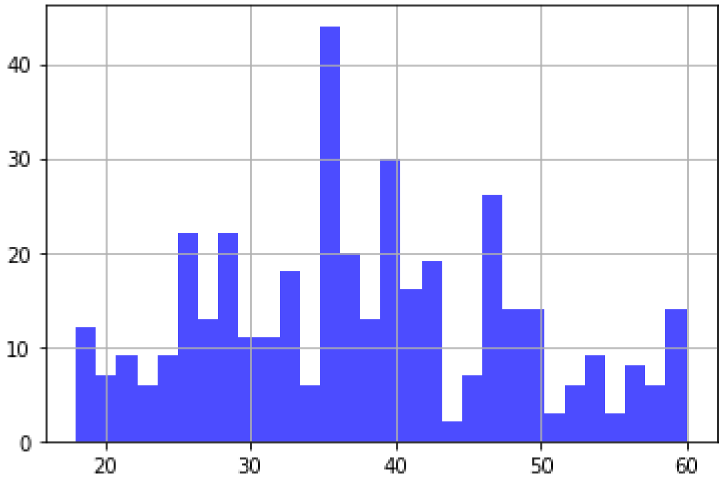
```
#Write the code to show the result(below)
```

In [51]:

```
data['Age'].hist(bins=30,color='blue',alpha=0.7)
```

Out[51]:

```
<AxesSubplot:>
```

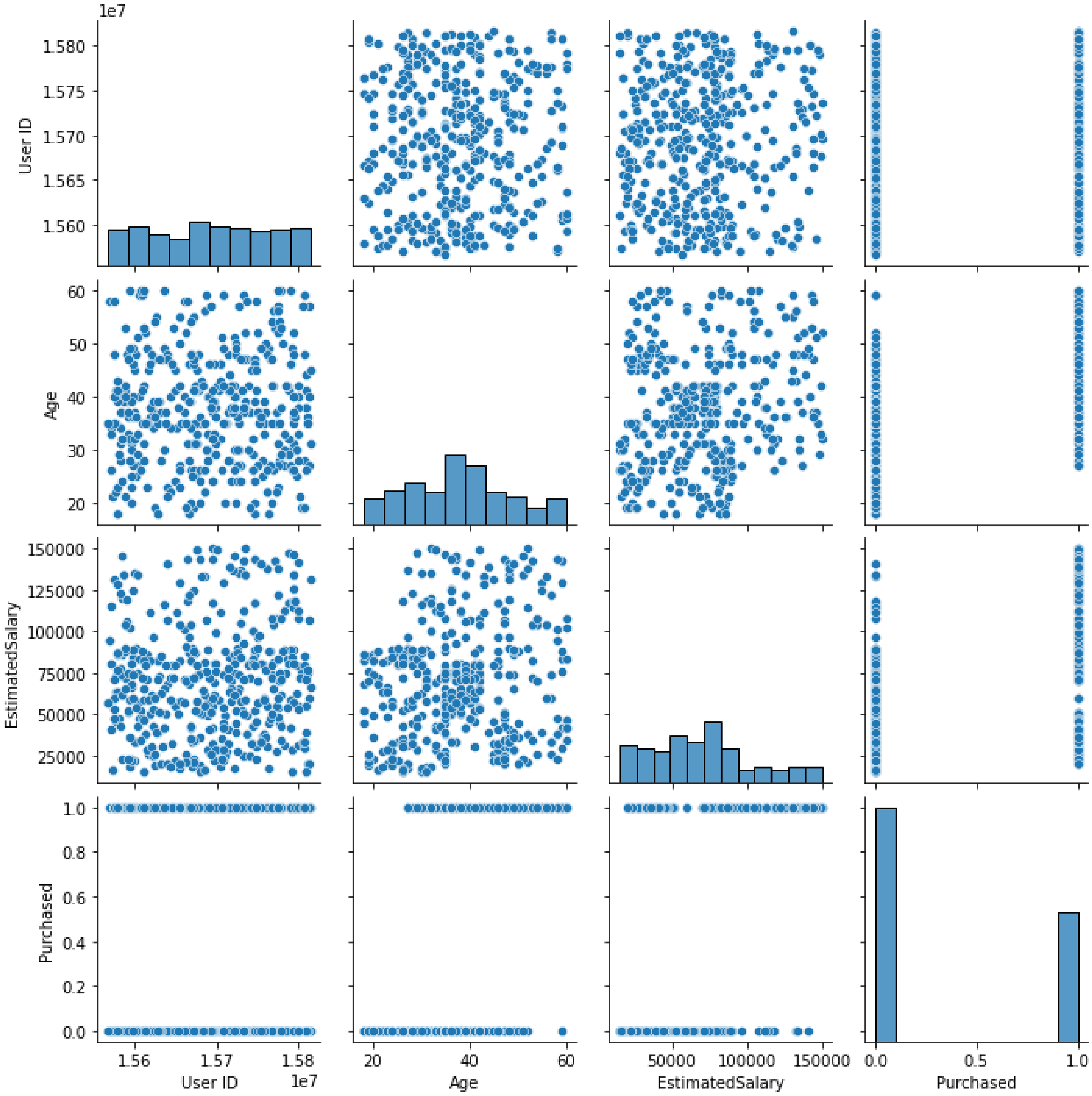


In [52]:

```
#Write the code to show the result(below)
sns.pairplot(data)
```

Out[52]:

```
<seaborn.axisgrid.PairGrid at 0x2651bb107f0>
```



Logistic Regression

Now it's time to do a train test split, and train our model. Check for categorical and Numerical values and buid the model based on your data type.

In [53]:

```
from sklearn.model_selection import train_test_split


gender = pd.get_dummies(data['Gender'],drop_first=True)
data.drop(['Gender'],axis=1,inplace=True)
data = pd.concat([data, gender],axis=1)
```

In [55]:

```
data.head()
```

Out[55]:

	User ID	Age	EstimatedSalary	Purchased	Male
0	15624510	19	19000	0	1
1	15810944	35	20000	0	1
2	15668575	26	43000	0	0
3	15603246	27	57000	0	0
4	15804002	19	76000	0	1

In [67]:

```
x = data[['Age', 'EstimatedSalary', 'Male']]
y = data['Purchased']

X_train, X_test, y_train, y_test = train_test_split( x,y, test_size=0.30,random_state=0)
```

In [68]:

```
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

Out[68]:

```
LogisticRegression()
```

Predictions and Evaluations

In [69]:

```
predictions = logmodel.predict(X_test)
```

In [70]:

```
predictions
```

Out[70]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

Classification Report

In [71]:

```
from sklearn.metrics import classification_report
```

In [72]:

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.66      1.00      0.79        79
           1       0.00      0.00      0.00        41

    accuracy                           0.66       120
   macro avg       0.33      0.50      0.40       120
weighted avg       0.43      0.66      0.52       120


C:\Users\meet2\anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1245: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\meet2\anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1245: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\meet2\anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1245: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: