PROJECT

PROBLEM STATEMENT :

Ecommerce company based in New York City that sells clothing online but they also have in-store style and clothing advice sessions. Customers come in to the store, have sessions/meetings with a personal stylist, then they can go home and order either on a mobile app or website for the clothes they want.

The company is trying to decide whether to focus their efforts on their mobile app experience or their website.

DATASET SOURCE : Kaggle.com

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [6]:
```
df = pd.read_csv("Ecommerce Customers")
```

In [8]:
```
df.head()
```

Out[8]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	4.082621	587.951054
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	2.664034	392.204933
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	4.104543	487.547505
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	581.852344
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.446308	599.406092

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    object 
 2   Avatar                500 non-null    object 
 3   Avg. Session Length   500 non-null    float64
 4   Time on App           500 non-null    float64
 5   Time on Website       500 non-null    float64
 6   Length of Membership  500 non-null    float64
 7   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

In [10]:
```
df.describe()
```

Out[10]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
25%	32.341822	11.388153	36.349257	2.930450	445.038277
50%	33.082008	11.983231	37.069367	3.533975	498.887875
75%	33.711985	12.753850	37.716432	4.126502	549.313828
max	36.139662	15.126994	40.005182	6.922689	765.518462

In [12]:
```
df.columns
```

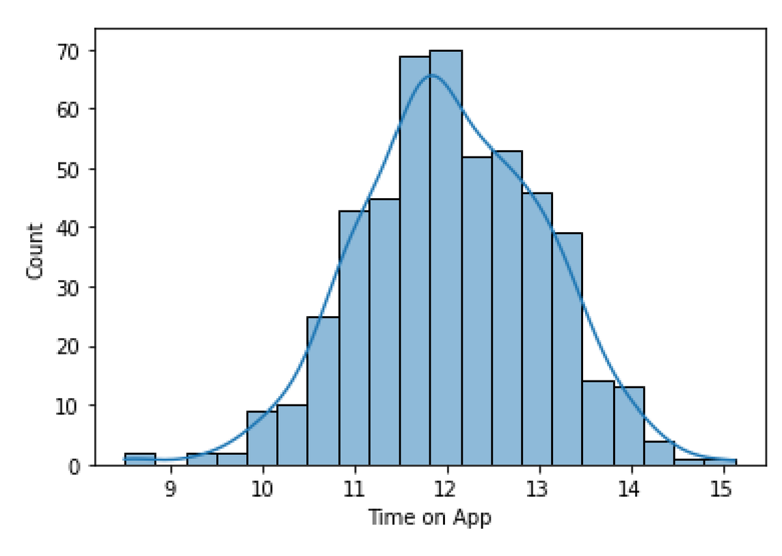
Out[12]:
```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

Exploratory Data Analysis

For the rest of the exercise we'll only be using the numerical data of the csv file

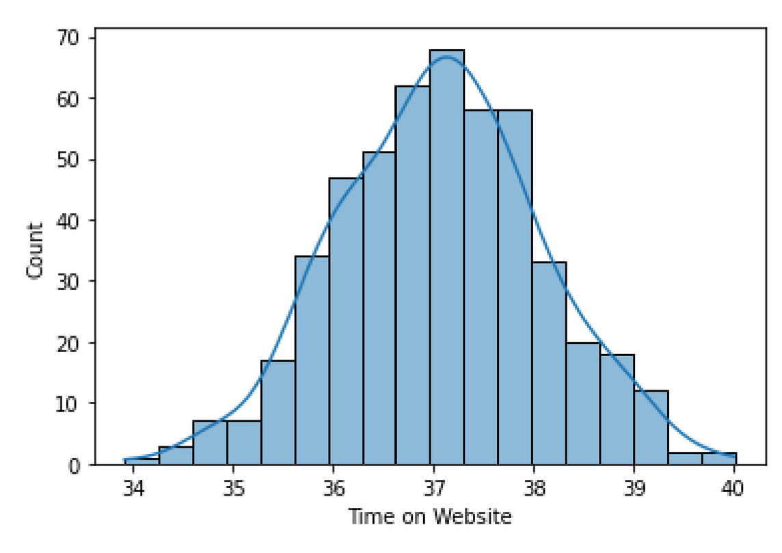
In [15]:
```
sns.histplot(df['Time on App'],kde = True)
```

Out[15]:
```
<AxesSubplot:xlabel='Time on App', ylabel='Count'>
```



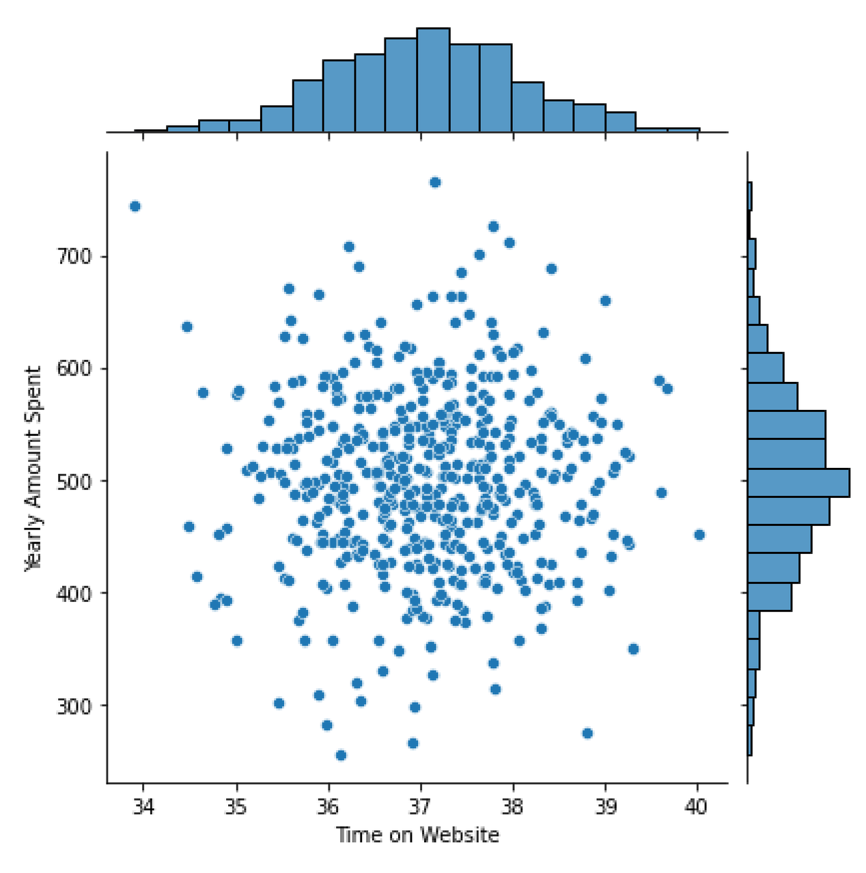
In [16]:
```
sns.histplot(df['Time on Website'],kde = True)
```

Out[16]:
```
<AxesSubplot:xlabel='Time on Website', ylabel='Count'>
```



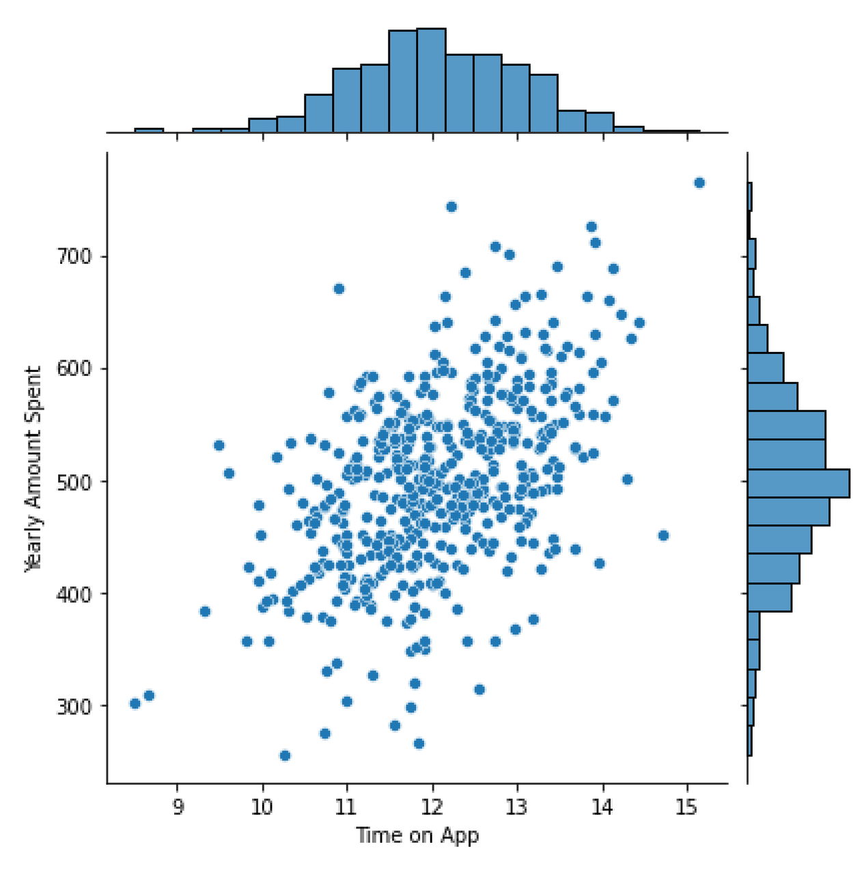
In [18]:
```
sns.jointplot(x=df['Time on Website'],y=df['Yearly Amount Spent'])
```

Out[18]:
```
<seaborn.axisgrid.JointGrid at 0x18da7337670>
```



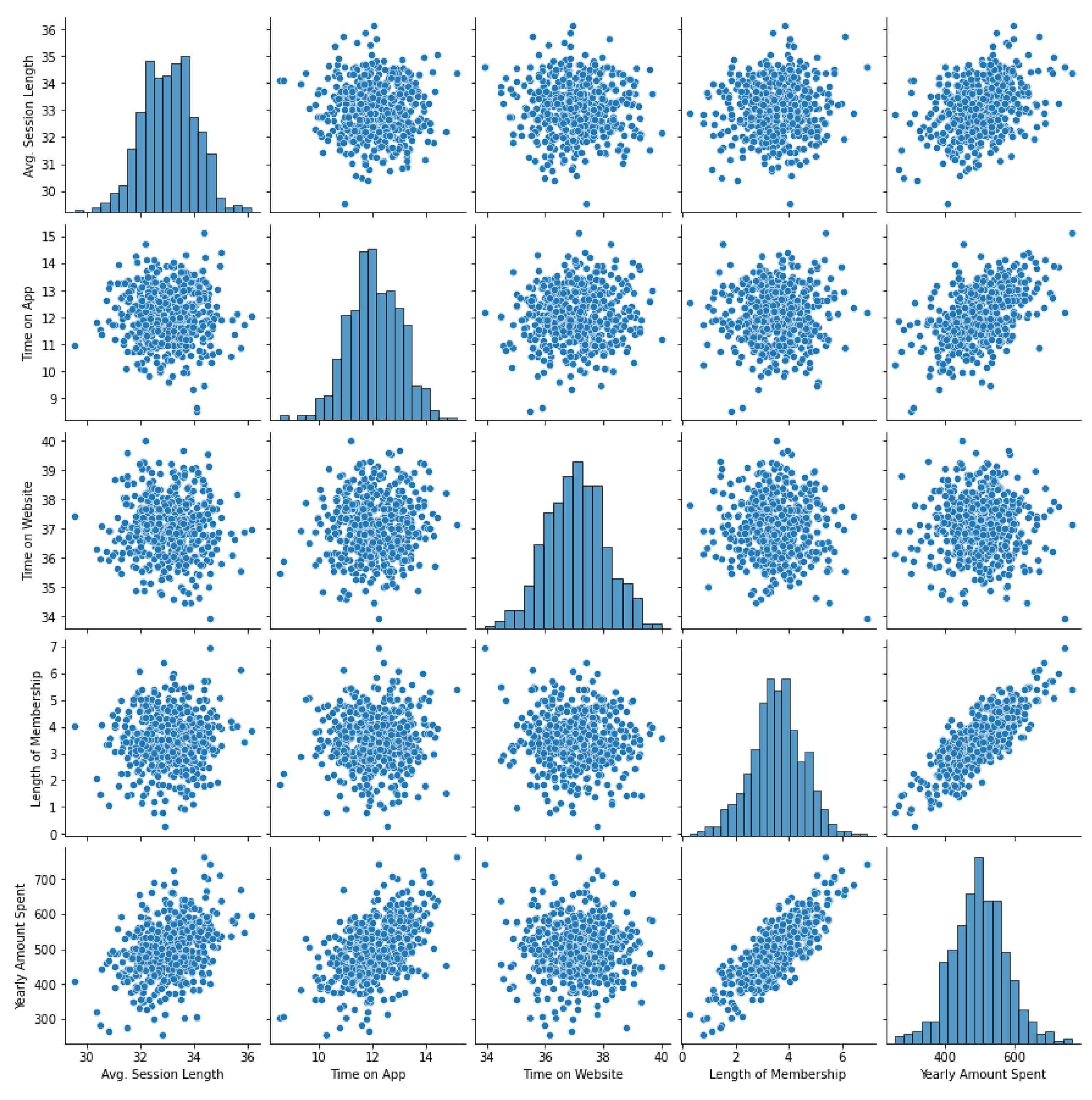
In [19]:
```
sns.jointplot(x=df['Time on App'],y=df['Yearly Amount Spent'])
```

Out[19]:
```
<seaborn.axisgrid.JointGrid at 0x18da78fa520>
```



In [20]:
```
sns.pairplot(df)
```

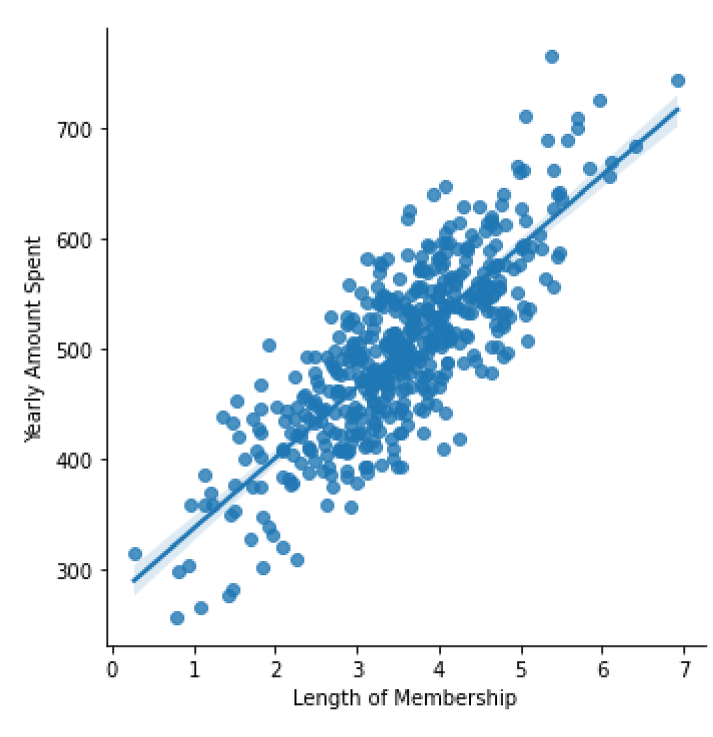
Out[20]:
```
<seaborn.axisgrid.PairGrid at 0x18da78fa430>
```



Based off this plot "LENGTH OF MEMBERSHIPS" looks to be the most correlated feature with "YEARLY AMOUNT SPENT".

In [24]:
```
sns.lmplot(x='Length of Membership',y='Yearly Amount Spent',data = df)
```

Out[24]:
```
<seaborn.axisgrid.FacetGrid at 0x18da983f3d0>
```



Implimentation of LinearRegression Model

In [26]:
```
y = df['Yearly Amount Spent']
X = df[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Membership']]
```

In [28]:
```
from sklearn.model_selection import train_test_split
```

In [29]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [ ]:

In [30]:
```
from sklearn.linear_model import LinearRegression
```

In [31]:
```
lm = LinearRegression()
```

In [32]:
```
lm.fit(X_train,y_train)
```

Out[32]:
```
LinearRegression()
```

Print out the coefficients of the model

In [33]:
```
print('Coefficients: \n', lm.coef_)
```

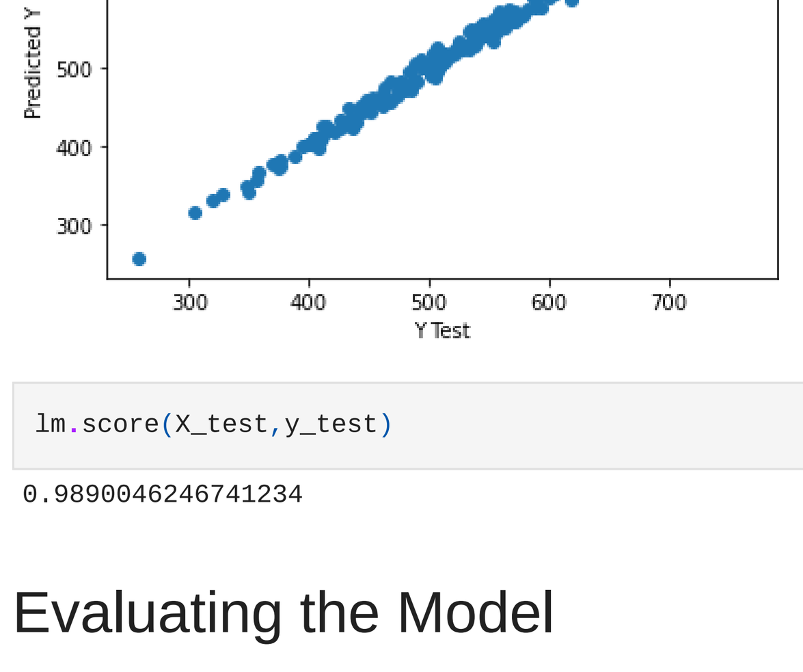
```
Coefficients: 
 [25.98154972 38.59015875  0.19040528 61.27909654]
```

In [34]:
```
predictions = lm.predict( X_test)
```

Create a scatterplot of the real test values versus the predicted values

In [35]:
```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[35]:
```
Text(0, 0.5, 'Predicted Y')
```



In [36]:
```
lm.score(X_test,y_test)
```

Out[36]:
```
0.9890046246741234
```

Evaluating the Model

```
Let's evaluate our model performance by calculating the residual sum of squares and the explained variance score (R^2).

**Calculate the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error. Refer to the lecture or to Wikipedia for the
formulas**
```

In [37]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 7.22814865343083
MSE: 79.8130516509745
RMSE: 8.933815066978637
```

Recreate the dataframe below

In [38]:
```
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[38]:

	Coeffecient
Avg. Session Length	25.981550
Time on App	38.590159
Time on Website	0.190405
Length of Membership	61.279097

conclusion

**Interpreting the coefficients:

1. Holding all other features fixed, a 1 unit increase in Avg. Session Length is associated with an increase of 25.98 total dollars spent.
2. Holding all other features fixed, a 1 unit increase in Time on App is associated with an increase of 38.59 total dollars spent.
3. Holding all other features fixed, a 1 unit increase in Time on Website is associated with an increase of 0.19 total dollars spent.
4. Holding all other features fixed, a 1 unit increase in Length of Membership is associated with an increase of 61.27 total dollars spent.

There is two ways to think about it : Develop the Website to catch up to the performance of the mobile app, or develop the app more since that is what is working better. This sort of answer really depends on the other factors going on at the company, you would probably want to explore the relationship between Length of Membership and the App or the Website before coming to a conclusion!

```
**THANK YOU**
```