In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
df = pd.read_csv("online_shoppers_intention.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageVal
0	0	0.0	0	0.0	1	0.000000	0.20	0.20	
1	0	0.0	0	0.0	2	64.000000	0.00	0.10	
2	0	0.0	0	0.0	1	0.000000	0.20	0.20	
3	0	0.0	0	0.0	2	2.666667	0.05	0.14	
4	0	0.0	0	0.0	10	627.500000	0.02	0.05	

In [4]:

```
df.isnull().any()
```

Out[4]:

```
Administrative             False
Administrative_Duration    False
Informational              False
Informational_Duration     False
ProductRelated             False
ProductRelated_Duration    False
BounceRates                False
ExitRates                  False
PageValues                 False
SpecialDay                 False
Month                      False
OperatingSystems           False
Browser                    False
Region                     False
TrafficType                False
VisitorType                False
Weekend                    False
Revenue                    False
dtype: bool
```

In [5]:

```
revenue_df=pd.get_dummies(df['Revenue'],drop_first=True)
revenue_df
```

Out[5]:

	True
0	0
1	0
2	0
3	0
4	0
...	...
12325	0
12326	0
12327	0
12328	0
12329	0

12330 rows × 1 columns

In [6]:

```
print(revenue_df)
```

```
       True
0         0
1         0
2         0
3         0
4         0
...     ...
12325     0
12326     0
12327     0
12328     0
12329     0

[12330 rows x 1 columns]
```

In [7]:

```
df.drop(['Revenue'],axis=1,inplace=True)
df
```

Out[7]:

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	Pag
0	0	0.0	0	0.0	1	0.000000	0.200000	0.200000	0
1	0	0.0	0	0.0	2	64.000000	0.000000	0.100000	0
2	0	0.0	0	0.0	1	0.000000	0.200000	0.200000	0
3	0	0.0	0	0.0	2	2.666667	0.050000	0.140000	0
4	0	0.0	0	0.0	10	627.500000	0.020000	0.050000	0
...	...	...	...	...	...	...	...	...	
12325	3	145.0	0	0.0	53	1783.791667	0.007143	0.029031	12
12326	0	0.0	0	0.0	5	465.750000	0.000000	0.021333	0
12327	0	0.0	0	0.0	6	184.250000	0.083333	0.086667	0
12328	4	75.0	0	0.0	15	346.000000	0.000000	0.021053	0
12329	0	0.0	0	0.0	3	21.250000	0.000000	0.066667	0

12330 rows × 17 columns

In [8]:

```
df=pd.concat([df,revenue_df],axis=1)
df
```

Out[8]:

ductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	VisitorType	Weekend
0.000000	0.200000	0.200000	0.000000	0.0	Feb	1	1	1	1	Returning_Visitor	False
64.000000	0.000000	0.100000	0.000000	0.0	Feb	2	2	1	2	Returning_Visitor	False
0.000000	0.200000	0.200000	0.000000	0.0	Feb	4	1	9	3	Returning_Visitor	False
2.666667	0.050000	0.140000	0.000000	0.0	Feb	3	2	2	4	Returning_Visitor	False
627.500000	0.020000	0.050000	0.000000	0.0	Feb	3	3	1	4	Returning_Visitor	True
...	...	...	...	...	...	...	...	...	...	...	...
1783.791667	0.007143	0.029031	12.241717	0.0	Dec	4	6	1	1	Returning_Visitor	True
465.750000	0.000000	0.021333	0.000000	0.0	Nov	3	2	1	8	Returning_Visitor	True
184.250000	0.083333	0.086667	0.000000	0.0	Nov	3	2	1	13	Returning_Visitor	True
346.000000	0.000000	0.021053	0.000000	0.0	Nov	2	2	3	11	Returning_Visitor	False
21.250000	0.000000	0.066667	0.000000	0.0	Nov	3	2	1	2	New_Visitor	True

In [9]:

```
X = df.iloc[:, [5, 6,7]].values
X
```

Out[9]:

```
array([[0.0000000e+00, 2.0000000e-01, 2.0000000e-01],
       [6.4000000e+01, 0.0000000e+00, 1.0000000e-01],
       [0.0000000e+00, 2.0000000e-01, 2.0000000e-01],
       ...,
       [1.8425000e+02, 8.3333333e-02, 8.6666667e-02],
       [3.4600000e+02, 0.0000000e+00, 2.1052632e-02],
       [2.1250000e+01, 0.0000000e+00, 6.6666667e-02]])
```

In [10]:

```
y = df.iloc[:, -1].values
y
```

Out[10]:

```
array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)
```

In [11]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

In [12]:

```
# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[12]:

```
▾ GaussianNB
GaussianNB()
```

In [16]:

```
y_pred = classifier.predict(X_test)
print('Predicted Value')
print(y_pred[:5])
print('Actual Value')
print(y_test[:5])
```

```
Predicted Value
[0 1 0 0 0]
Actual Value
[0 0 0 0 0]
```

In [14]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
ac = accuracy_score(y_test,y_pred)
ac
```

Out[14]:

```
0.8203568532035685
```

In [21]:

```
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[21]:

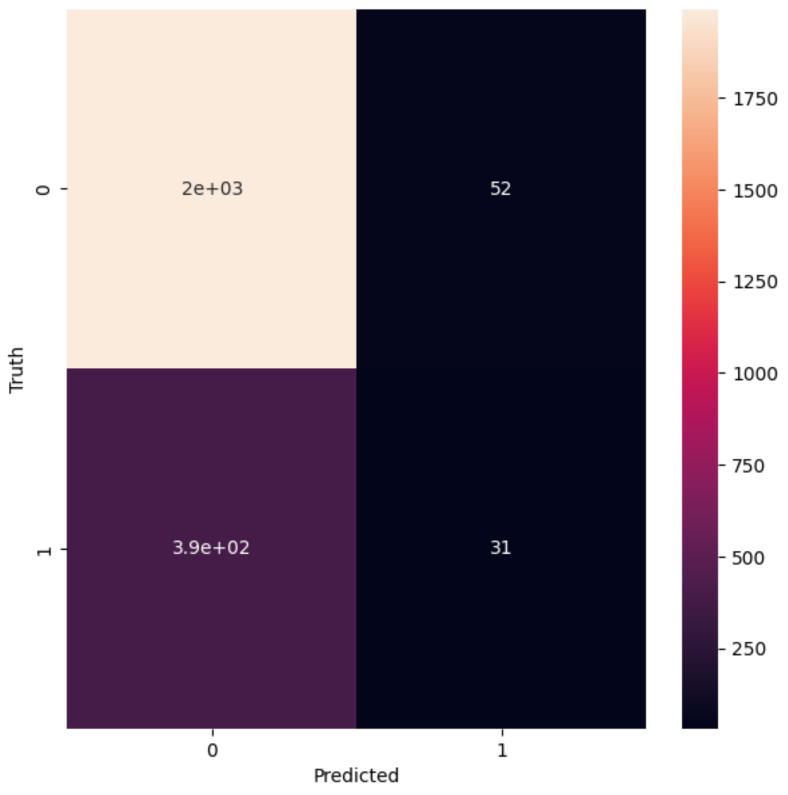
```
array([[1992,   52],
       [ 391,   31]])
```

In [22]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(7,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[22]:

```
Text(58.222222222222214, 0.5, 'Truth')
```



In [ ]: