In [91]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import sklearn
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [8]:

```
df = pd.read_csv("Admission_Predict_Ver1.1.csv")
```

In [13]:

```
df.shape
```

Out[13]:

```
(500, 9)
```

In [19]:

```
df.head()
```

Out[19]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [20]:

```
df.tail()
```

Out[20]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
495	496	332	108	5	4.5	4.0	9.02	1	0.87
496	497	337	117	5	5.0	5.0	9.87	1	0.96
497	498	330	120	5	4.5	5.0	9.56	1	0.93
498	499	312	103	4	4.0	5.0	8.43	0	0.73
499	500	327	113	4	4.5	4.5	9.04	0	0.84

In [27]:

```
#Normalizing data
df['GRE Score'] = df['GRE Score']/380
df['University Rating'] = df['University Rating']/5
df['SOP']=df['SOP']/5
df['LOR ']=df['LOR ']/5
df['CGPA']=df['CGPA']/10
df['TOEFL Score']=df['TOEFL Score']/120
```

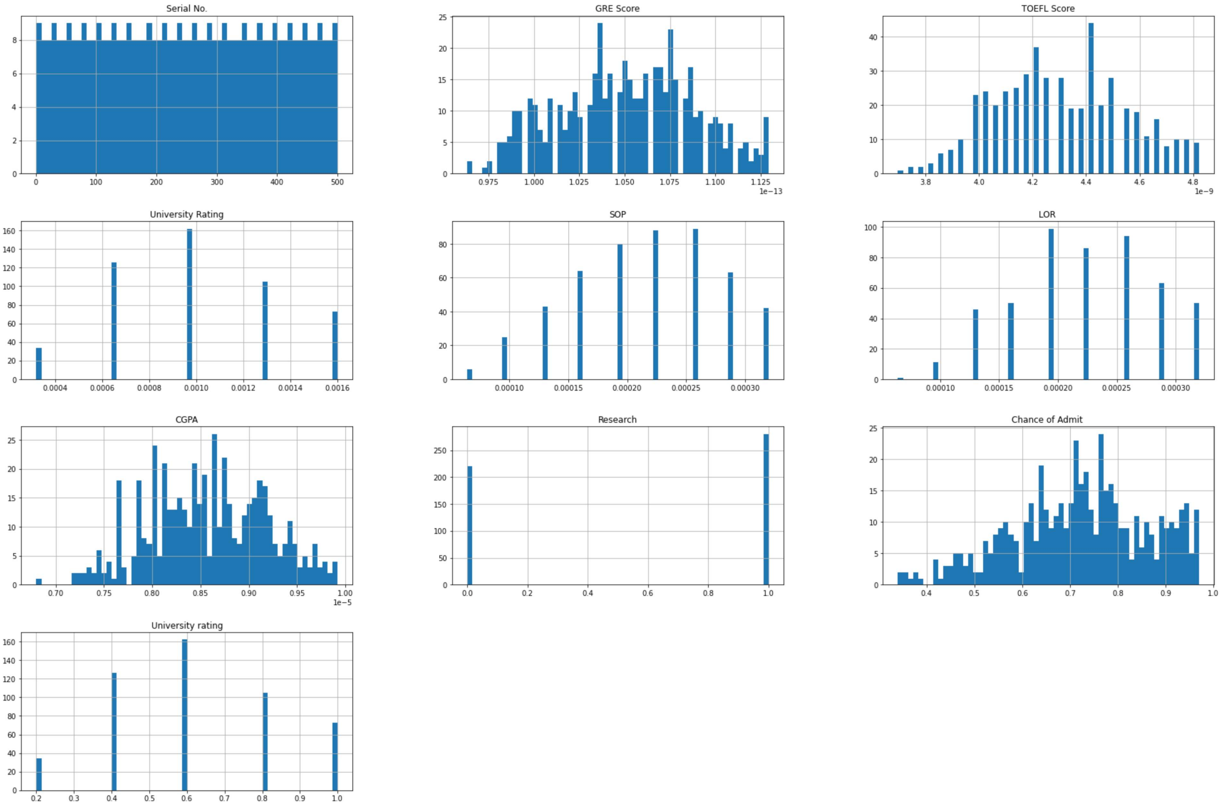
In [35]:

```
#Data Visualization
df.hist(bins=60, figsize=(30,20))
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-35-5b988b3067df> in <module>
      1 df.hist(bins=60, figsize=(30,20))
```

```
----> 2 plt.show()

AttributeError: module 'matplotlib' has no attribute 'show'
```



In [36]:

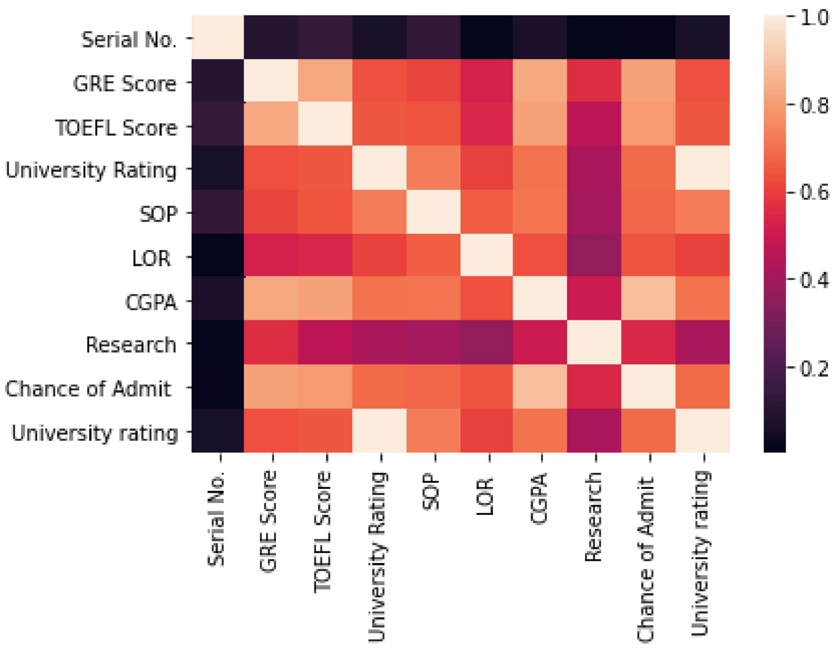
```
#Checking the coorelation between values
corr_matrix= df.corr().abs()
```

In [37]:

```
sns.heatmap(corr_matrix)
```

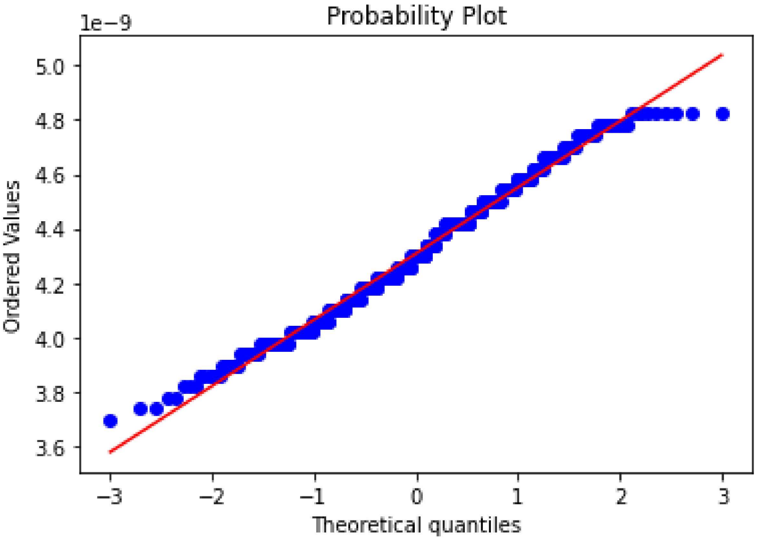
Out[37]:

```
<AxesSubplot:>
```



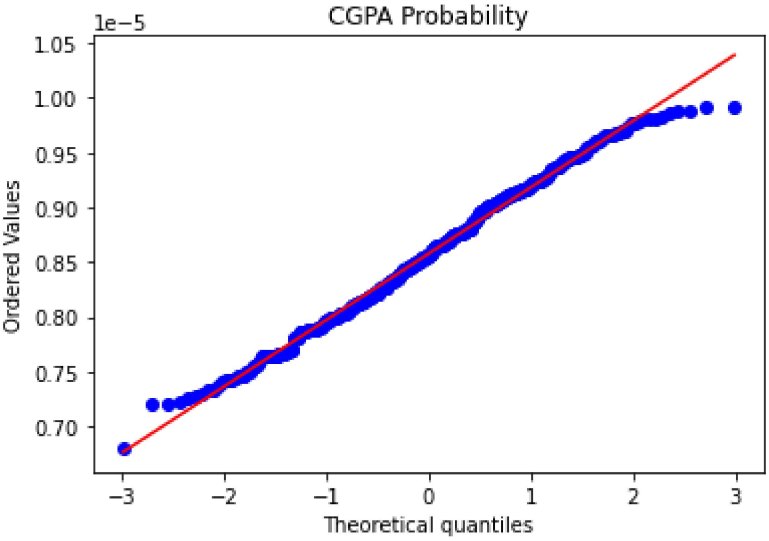
In [48]:

```
#Checking Normal Distribution of data for Linear Regression
nd = stats.probplot(df['TOEFL Score'], plot=plt)
plt.show()
```



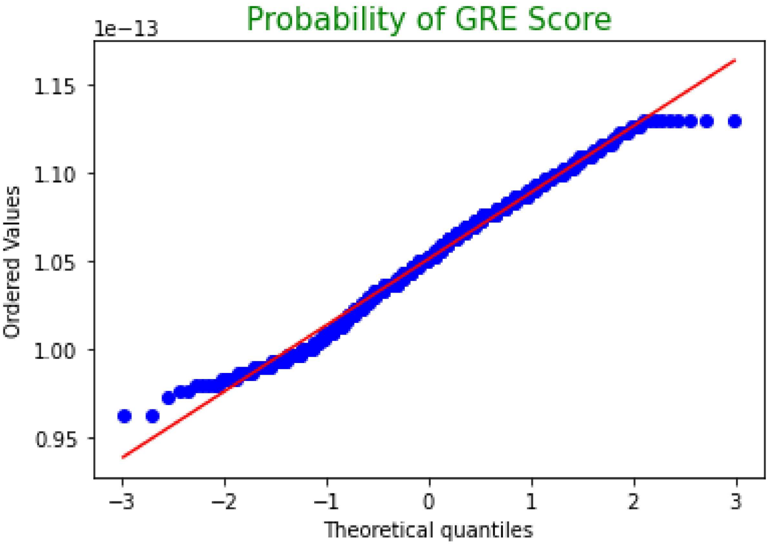
```
In [54]: nd = stats.probplot(df['CGPA'], plot=plt)
         plt.title('CGPA Probability')
```

```
Out[54]: Text(0.5, 1.0, 'CGPA Probability')
```



```
In [63]: nd = stats.probplot(df['GRE Score'], plot=plt)
         plt.title('Probability of GRE Score', color='Green', size=15)
```

```
Out[63]: Text(0.5, 1.0, 'Probability of GRE Score')
```



In [74]:

```
#Predicting Likelihood of Admission
y_df = df['Chance of Admit ']
#df.drop('Serial No.', inplace = True)
#df.drop('Chance of Admit ', inplace=True)
```

In [106…

```
X_train, X_test, y_train, y_test = train_test_split(df, y_df, test_size=0.2)
lm = linear_model.LinearRegression()
lm.fit(X_train,y_train)
```

Out[106…

```
LinearRegression()
```

In [111…

```
r = lm.score((X_test, y_test)* 100)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-111-ffa903212703> in <module>
----> 1 r = lm.score((X_test, y_test)* 100)

TypeError: score() missing 1 required positional argument: 'y'
```

In [ ]:

```

```