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

In [2]:
```
df=pd.read_csv(r"C:\Users\Admin\Downloads\13_placement - 13_placement.csv")
df
```

Out[2]:

	cgpa	placement_exam_marks	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1
3	6.42	8	1
4	7.23	17	0
...	...	...	...
995	8.87	44	1
996	9.12	65	1
997	4.89	34	0
998	8.62	46	1
999	4.90	10	1

1000 rows × 3 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   cgpa                 1000 non-null   float64
 1   placement_exam_marks  1000 non-null   int64  
 2   placed               1000 non-null   int64  
dtypes: float64(1), int64(2)
memory usage: 23.6 KB
```

In [4]:
```
df.info()
```

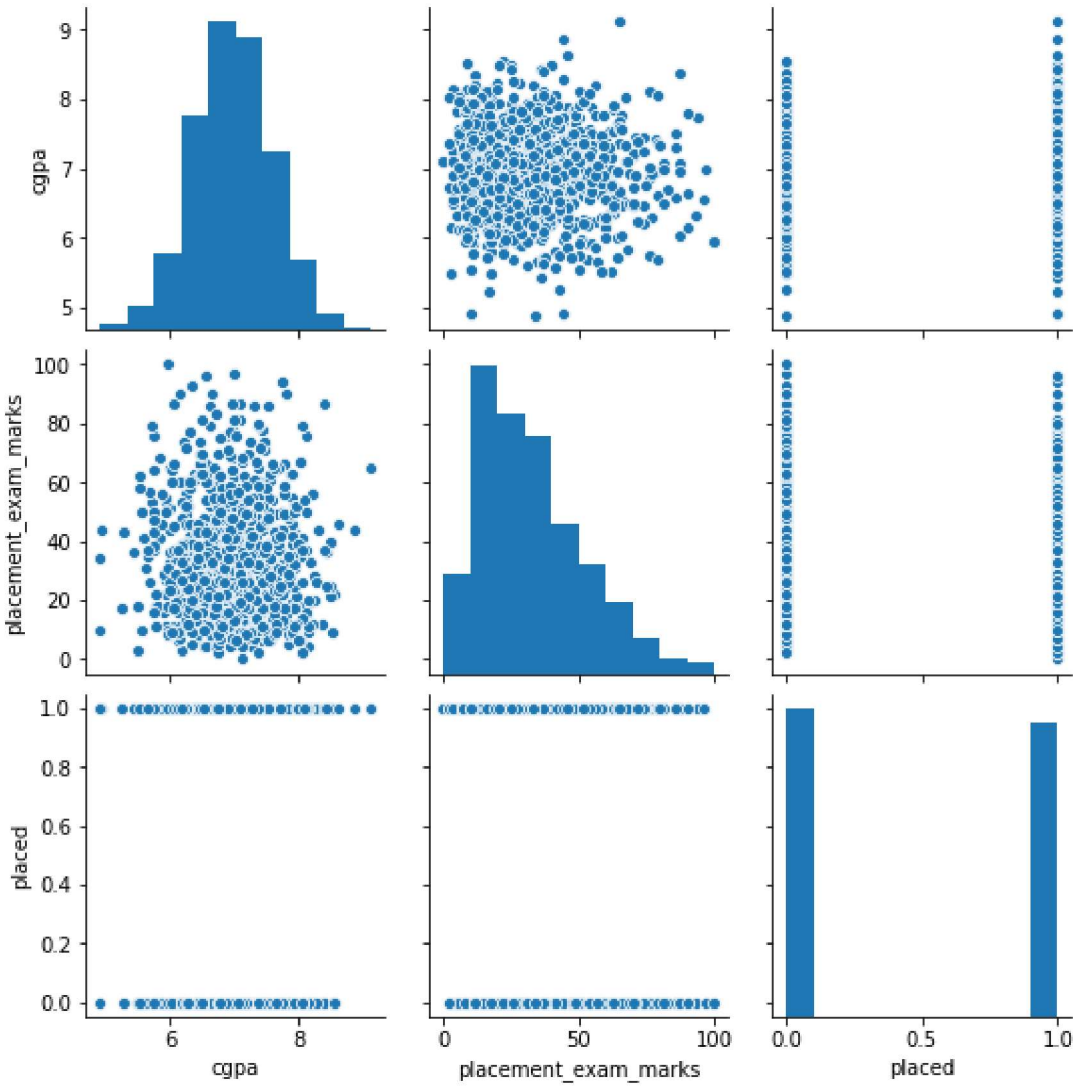
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   cgpa                  1000 non-null   float64
 1   placement_exam_marks  1000 non-null   int64  
 2   placed                1000 non-null   int64  
dtypes: float64(1), int64(2)
memory usage: 23.6 KB
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

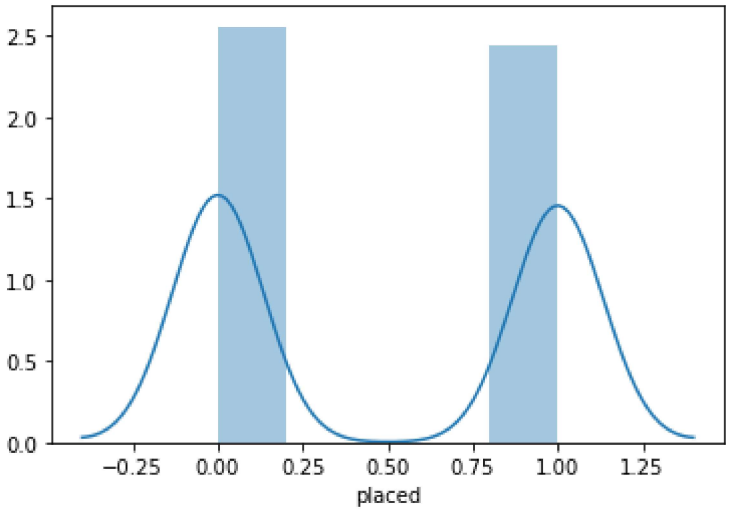
In [6]:
```
sns.pairplot(df)
```

Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x1466c2aba90>
```



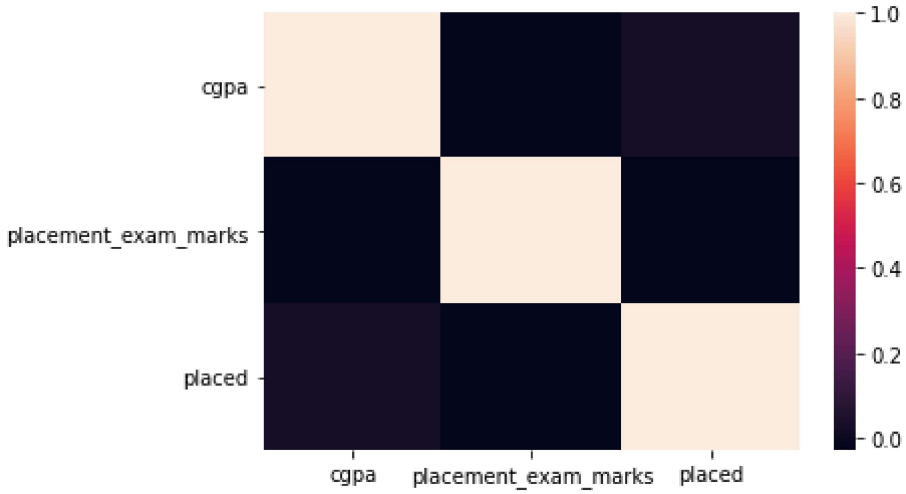
In [7]:
```
sns.distplot(df['placed'])
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1466c7c7610>
```



In [8]:
```
sns.heatmap(df.corr())
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1466ca17550>
```



In [9]:
```
x=df[['cgpa', 'placement_exam_marks']]
y=df[['placed']]
```

In [10]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [11]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

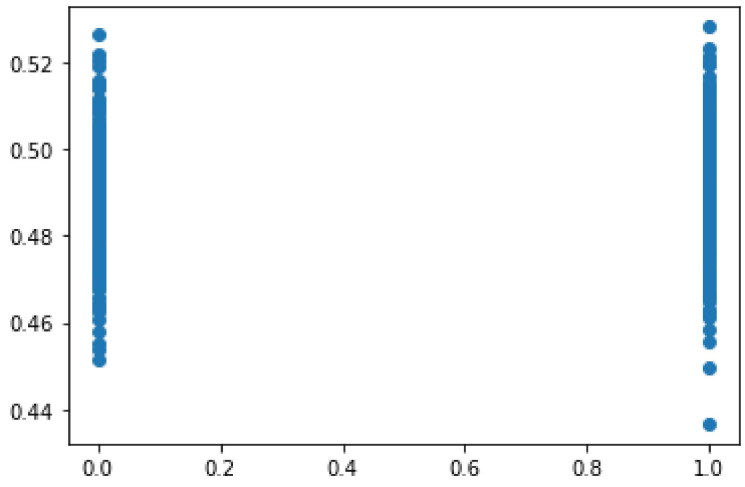
Out[11]:
```
LinearRegression()
```

In [12]:
```
print(lr.intercept_)
```

```
[0.30944966]
```

In [13]:
```
prediction= lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[13]: <matplotlib.collections.PathCollection at 0x1466d4eef40>



In [14]:
```
print(lr.score(x_test,y_test))
```

```
-4.4889221095800735e-06
```

In [15]:
```
print(lr.score(x_train,y_train))
```

```
0.0010532126462761138
```

In [16]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [17]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[17]: Ridge(alpha=10)

In [18]:
```
rr.score(x_test,y_test)
```

Out[18]: 3.566958748035809e-05

In [19]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[19]: Lasso(alpha=10)

In [20]:
```
la.score(x_test,y_test)
```

Out[20]: -4.447607187318958e-05

In [ ]: