In [ ]:

```

```

In [122]:

```
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [123]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\13_placement.csv")
a
```

Out[123]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
998	8.62	46.0	1
999	4.90	10.0	1

1000 rows × 3 columns

In [124]:

```
a=a.head(10)
a
```

Out[124]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
5	7.30	23.0	1
6	6.69	11.0	0
7	7.12	39.0	1
8	6.45	38.0	0
9	7.75	94.0	1

In [125]:

```
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   cgpa                  10 non-null     float64
 1   placement_exam_marks  10 non-null     float64
 2   placed                10 non-null     int64  
dtypes: float64(2), int64(1)
memory usage: 368.0 bytes
```

In [126]:

```
# to display summary of statastic
a.describe()
```

Out[126]:

	cgpa	placement_exam_marks	placed
count	10.000000	10.000000	10.000000
mean	7.115000	33.400000	0.700000
std	0.454832	24.423122	0.483046
min	6.420000	8.000000	0.000000
25%	6.797500	18.500000	0.250000
50%	7.210000	32.000000	1.000000
75%	7.420000	38.750000	1.000000
max	7.750000	94.000000	1.000000

In [127]:

```
# to display colum heading
a.columns
```

Out[127]:

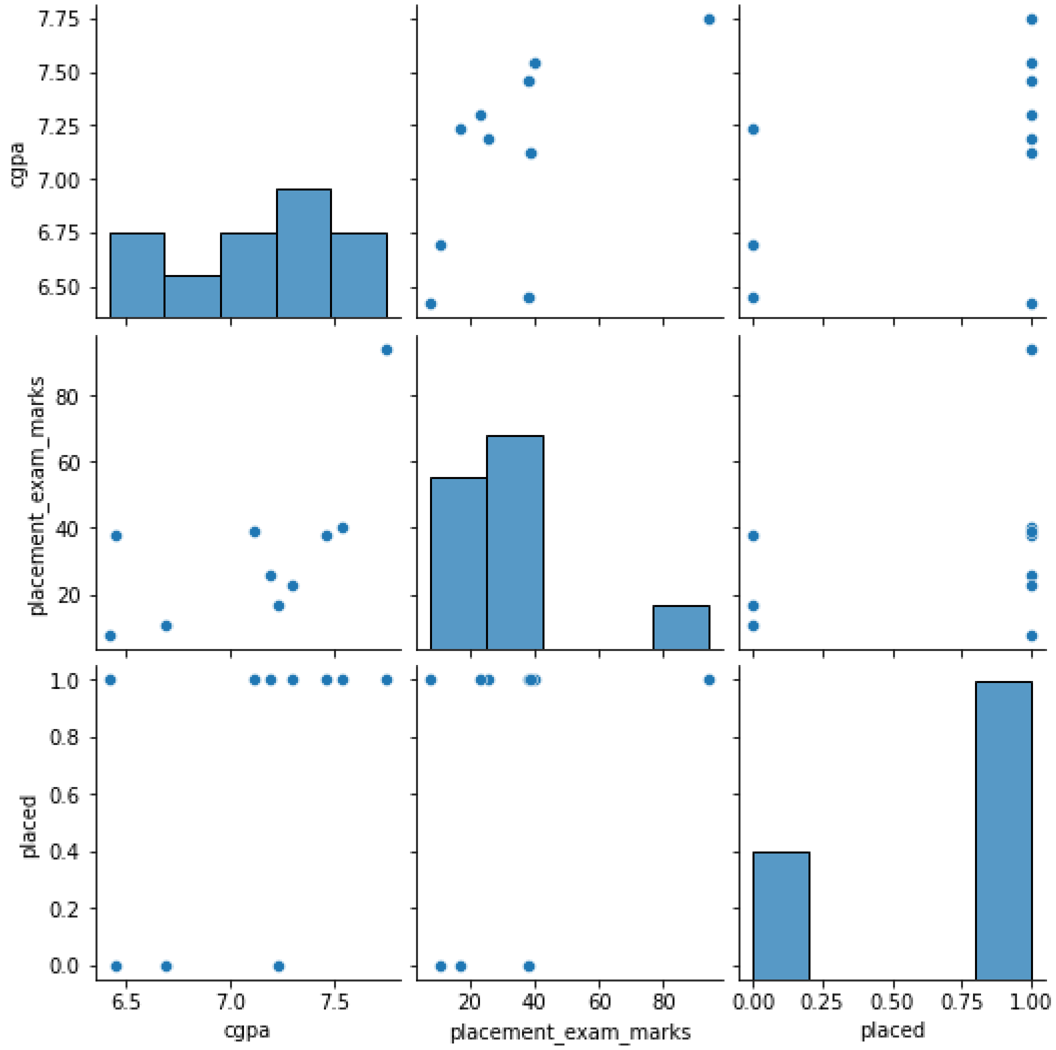
```
Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

In [128]:

```
sns.pairplot(a)
```

Out[128]:

```
<seaborn.axisgrid.PairGrid at 0x203e0d1ea60>
```

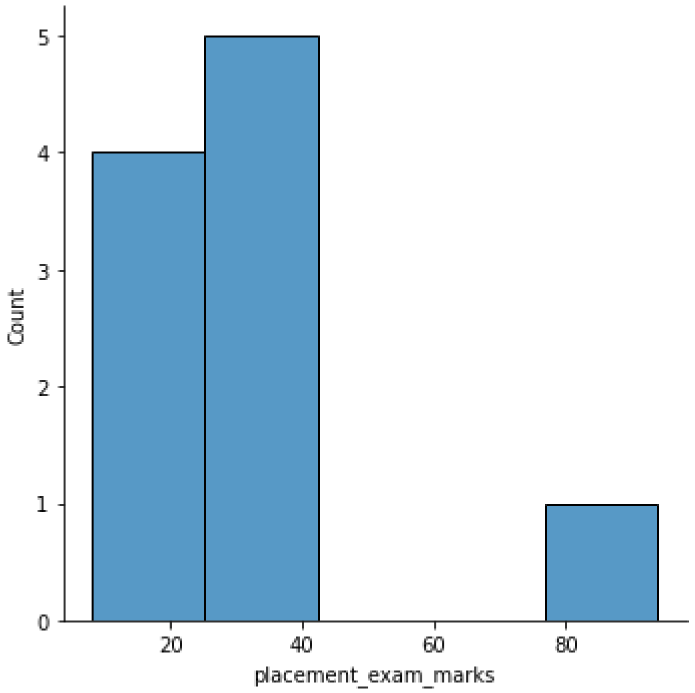


In [129]:

```
sns.displot(a["placement_exam_marks"])
```

Out[129]:

```
<seaborn.axisgrid.FacetGrid at 0x203df5b3d00>
```



In [130]:

```
b=a[['cgpa', 'placement_exam_marks', 'placed']]
b
```

Out[130]:

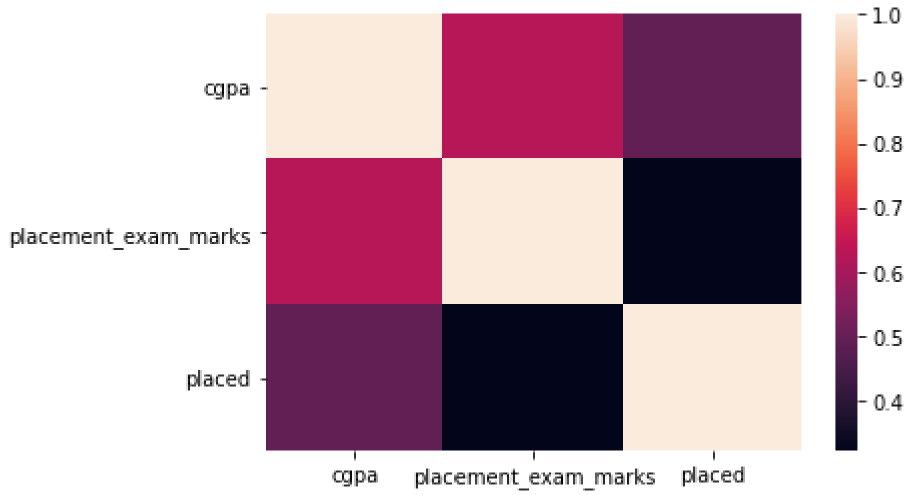
	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
5	7.30	23.0	1
6	6.69	11.0	0
7	7.12	39.0	1
8	6.45	38.0	0
9	7.75	94.0	1

In [131]:

```
sns.heatmap(b.corr())
```

Out[131]:

```
<AxesSubplot:>
```



In [133]:

```
x=a[['cgpa', 'placement_exam_marks', 'placed']]
y=a['placement_exam_marks']
```

In [134]:

```
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 [135]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[135]:

```
LinearRegression()
```

In [136]:

```
lr.intercept_
```

Out[136]:

```
-1.2079226507921703e-13
```

In [137]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[137]:

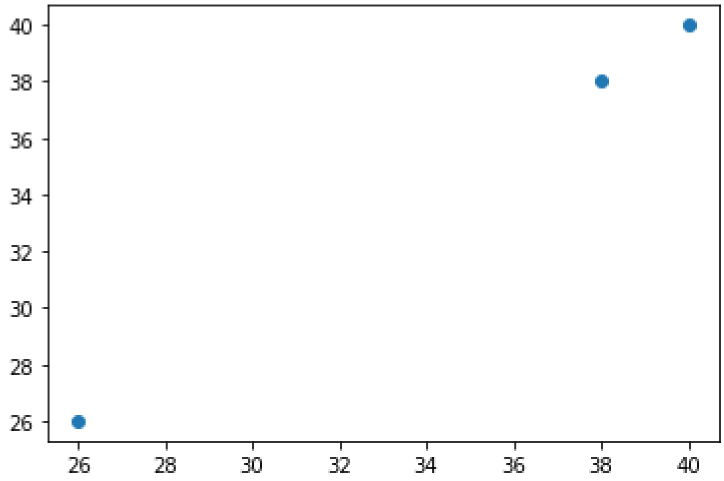
	Co-efficient
cgpa	1.911538e-14
placement_exam_marks	1.000000e+00
placed	8.441321e-15

In [138]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[138]:

```
<matplotlib.collections.PathCollection at 0x203e1519550>
```



In [139]:

```
lr.score(x_test,y_test)
```

Out[139]:

```
1.0
```

In [140]:

```
lr.score(x_train,y_train)
```

Out[140]:

```
1.0
```

In [141]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [142]:

```
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[142]:

```
Ridge(alpha=10)
```

In [143]:

```
rr.score(x_test,y_test)
```

Out[143]:

```
0.9935726121377305
```

In [144]:

```
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

Out[144]:

```
Lasso(alpha=10)
```

In [145]:

```
la.score(x_test,y_test)
```

Out[145]:

```
0.9315508382909682
```

In [ ]:

```

```