Import Library

In [46]:

```
import numpy as np
import pandas as pd
import seaborn as sns

from matplotlib import pyplot as plt
```

Loading Dataset

In [47]:

```
df = pd.read_csv('Placement_Data_Full_Class.csv')
df.head()
```

Out[47]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	et
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	

In [48]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           215 non-null    int64  
 1   gender          215 non-null    object 
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object 
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object 
 6   hsc_s           215 non-null    object 
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object 
 9   workex          215 non-null    object 
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object 
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

In [49]:

```
df.describe()
```

Out[49]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	148.000000
mean	108.000000	67.303395	66.333163	66.370186	72.100558	62.278186	288655.405405
std	62.209324	10.827205	10.897509	7.358743	13.275956	5.833385	93457.452420
min	1.000000	40.890000	37.000000	50.000000	50.000000	51.210000	200000.000000
25%	54.500000	60.600000	60.900000	61.000000	60.000000	57.945000	240000.000000
50%	108.000000	67.000000	65.000000	66.000000	71.000000	62.000000	265000.000000
75%	161.500000	75.700000	73.000000	72.000000	83.500000	66.255000	300000.000000
max	215.000000	89.400000	97.700000	91.000000	98.000000	77.890000	940000.000000

In [50]:

```
for i in df:
    if df[i].dtype == 'O':
        print(i,df[i].mode().values[0],'*'*80,sep='\n')
```

```
gender
M
*****************************************************************************
***
ssc_b
Central
*****************************************************************************
***
hsc_b
Others
*****************************************************************************
***
hsc_s
Commerce
*****************************************************************************
***
degree_t
Comm&Mgmt
*****************************************************************************
***
workex
No
*****************************************************************************
***
specialisation
Mkt&Fin
*****************************************************************************
***
status
Placed
*****************************************************************************
***
```

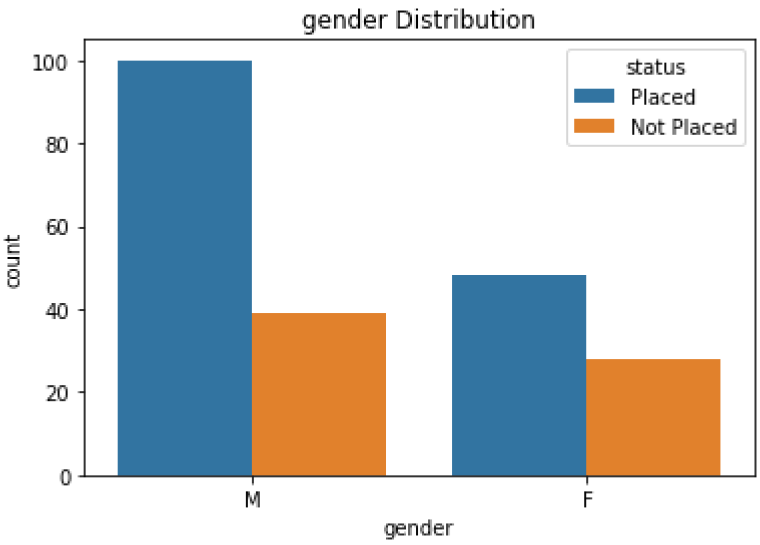
In [51]:
```
for i in df:
    if i != 'sl_no' and df[i].dtype == 'O':
        print(i,df[i].unique(),'*'*80,sep='\n')
```

```
gender
['M' 'F']
*****************************************************************************
***
ssc_b
['Others' 'Central']
*****************************************************************************
***
hsc_b
['Others' 'Central']
*****************************************************************************
***
hsc_s
['Commerce' 'Science' 'Arts']
*****************************************************************************
***
degree_t
['Sci&Tech' 'Comm&Mgmt' 'Others']
*****************************************************************************
***
workex
['No' 'Yes']
*****************************************************************************
***
specialisation
['Mkt&HR' 'Mkt&Fin']
*****************************************************************************
***
status
['Placed' 'Not Placed']
*****************************************************************************
***
```

EDA

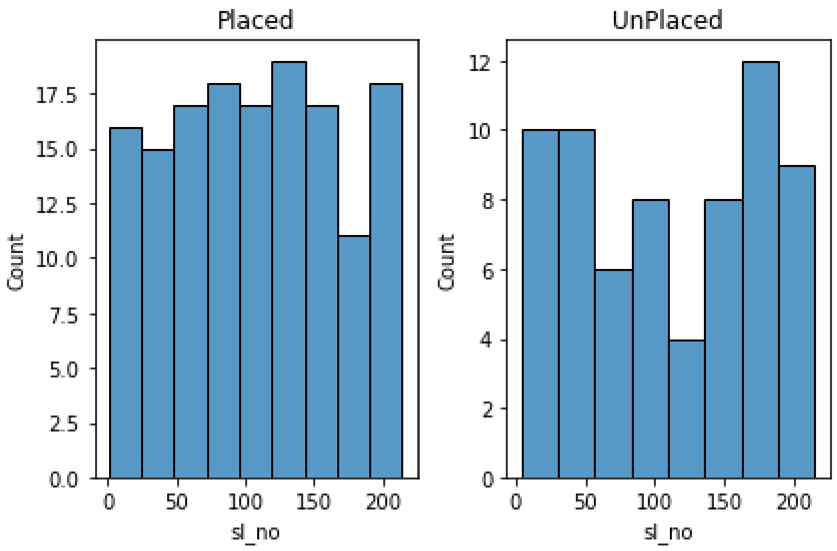
Distribution of Placed and Unplaced as per categorial feature

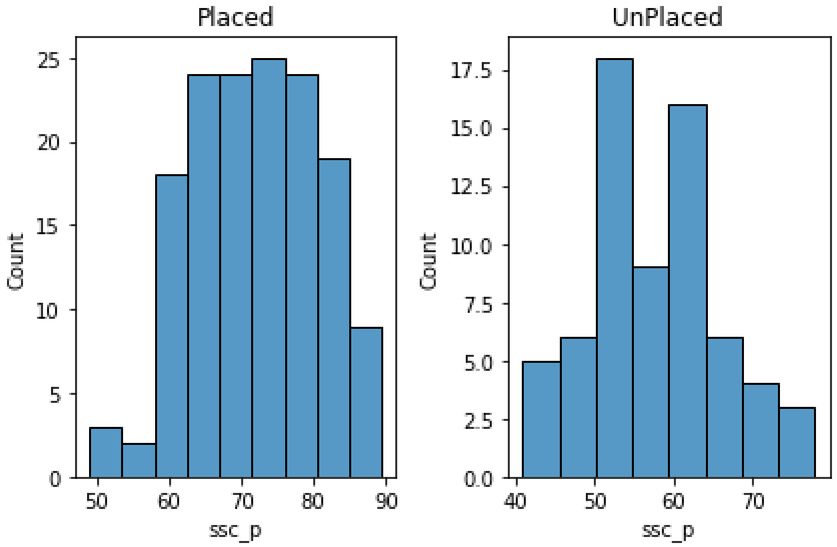
In [52]:
```
for i in df:
    if df[i].dtype == 'O' and i != 'status':
        sns.countplot(x = df[i], hue= df.status)
        plt.title(f"{i} Distribution")
        plt.show()
```

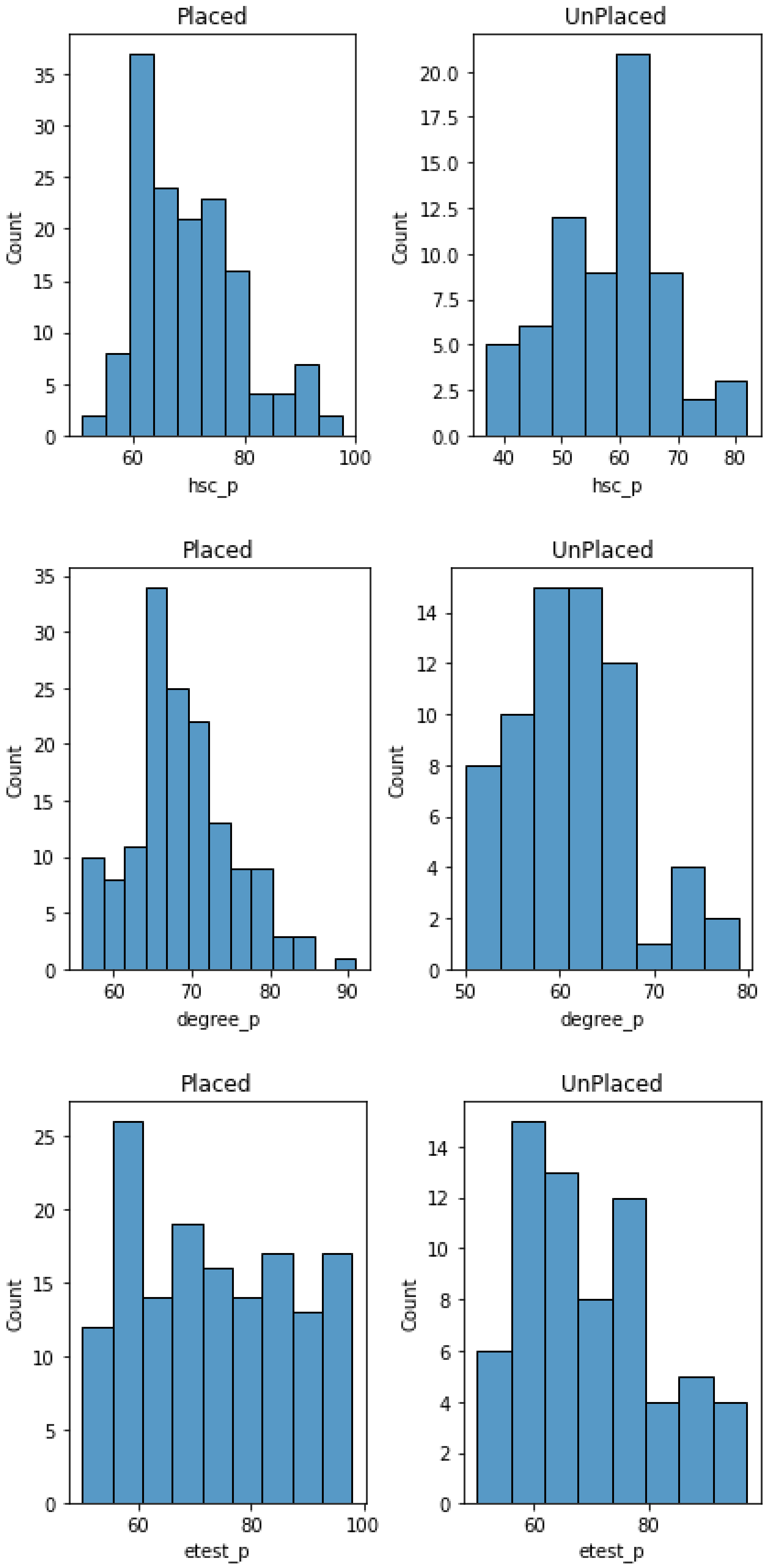


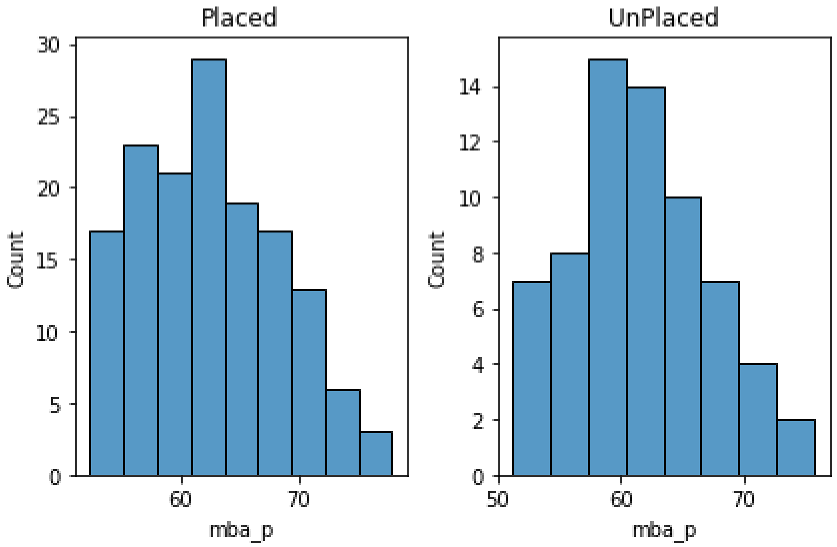
Distribution of Placed and Unplaced as per numerical feature

In [53]:
```
for i in df:
    if df[i].dtype != 'O' and i != 'salary':
        fig,axs = plt.subplots(1,2)

        sns.histplot(df[df['status']=='Placed'][i], ax = axs[0])
        axs[0].set_title("Placed")

        sns.histplot(df[df['status']!='Placed'][i], ax = axs[1])
        axs[1].set_title("UnPlaced")
        fig.tight_layout()
        plt.show()
```






Placed
Count
0
5
10
15
20
25
30
35
60
80
100
hsc_p
UnPlaced
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
40
50
60
70
80
hsc_p
Placed
Count
0
5
10
15
20
25
30
35
60
70
80
90
degree_p
UnPlaced
Count
0
2
4
6
8
10
12
14
50
60
70
80
degree_p
Placed
Count
0
5
10
15
20
25
60
80
100
etest_p
UnPlaced
Count
0
2
4
6
8
10
12
14
60
80
etest_p



Dataset of Placed students

In [54]:
```
placed = df[df["status"] == "Placed"]
placed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 148 entries, 0 to 213
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           148 non-null    int64  
 1   gender          148 non-null    object 
 2   ssc_p           148 non-null    float64
 3   ssc_b           148 non-null    object 
 4   hsc_p           148 non-null    float64
 5   hsc_b           148 non-null    object 
 6   hsc_s           148 non-null    object 
 7   degree_p        148 non-null    float64
 8   degree_t        148 non-null    object 
 9   workex          148 non-null    object 
 10  etest_p         148 non-null    float64
 11  specialisation  148 non-null    object 
 12  mba_p           148 non-null    float64
 13  status          148 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 18.5+ KB
```

In [55]:
```
placed.describe()
```

Out[55]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	148.000000	148.000000	148.000000	148.000000	148.000000	148.000000	148.000000
mean	106.878378	71.721486	69.926554	68.740541	73.238041	62.579392	288655.405405
std	60.682502	8.715445	9.329268	6.518087	13.729333	5.884583	93457.452420
min	1.000000	49.000000	50.830000	56.000000	50.000000	52.380000	200000.000000
25%	57.750000	65.000000	63.000000	65.000000	60.000000	57.772500	240000.000000
50%	108.500000	72.500000	68.000000	68.000000	72.000000	62.245000	265000.000000
75%	153.250000	78.125000	75.250000	72.422500	85.000000	66.760000	300000.000000
max	214.000000	89.400000	97.700000	91.000000	98.000000	77.890000	940000.000000

In [56]:
```
placed.describe(include='object')
```

Out[56]:

	gender	ssc_b	hsc_b	hsc_s	degree_t	workex	specialisation	status
count	148	148	148	148	148	148	148	148
unique	2	2	2	3	3	2	2	1
top	M	Central	Others	Commerce	Comm&Mgmt	No	Mkt&Fin	Placed
freq	100	78	91	79	102	84	95	148

Dataset of Unplaced Students

In [57]:

```
unplaced = df[df["status"] != "Placed"]
unplaced.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 67 entries, 3 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           67 non-null     int64  
 1   gender          67 non-null     object 
 2   ssc_p           67 non-null     float64
 3   ssc_b           67 non-null     object 
 4   hsc_p           67 non-null     float64
 5   hsc_b           67 non-null     object 
 6   hsc_s           67 non-null     object 
 7   degree_p        67 non-null     float64
 8   degree_t        67 non-null     object 
 9   workex          67 non-null     object 
 10  etest_p         67 non-null     float64
 11  specialisation  67 non-null     object 
 12  mba_p           67 non-null     float64
 13  status          67 non-null     object 
 14  salary          0 non-null      float64
dtypes: float64(6), int64(1), object(8)
memory usage: 8.4+ KB
```

In [58]:

```
unplaced.describe()
```

Out[58]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	67.000000	67.000000	67.000000	67.000000	67.000000	67.000000	0.0
mean	110.477612	57.544030	58.395522	61.134179	69.587910	61.612836	NaN
std	65.859667	8.394246	9.914090	6.365825	11.930687	5.705689	NaN
min	4.000000	40.890000	37.000000	50.000000	50.000000	51.210000	NaN
25%	48.500000	52.000000	51.000000	57.000000	60.000000	58.480000	NaN
50%	107.000000	56.280000	60.330000	61.000000	67.000000	60.690000	NaN
75%	170.500000	63.000000	64.000000	65.000000	76.500000	65.405000	NaN
max	215.000000	77.800000	82.000000	79.000000	97.000000	75.710000	NaN

In [59]:

```
unplaced.describe(include='object')
```

Out[59]:

	gender	ssc_b	hsc_b	hsc_s	degree_t	workex	specialisation	status
count	67	67	67	67	67	67	67	67
unique	2	2	2	3	3	2	2	1
top	M	Central	Others	Commerce	Comm&Mgmt	No	Mkt&HR	Not Placed
freq	39	38	40	34	43	57	42	67

Checking Mininum Eligibility Criteria of Getting Placed

SSC Percentage

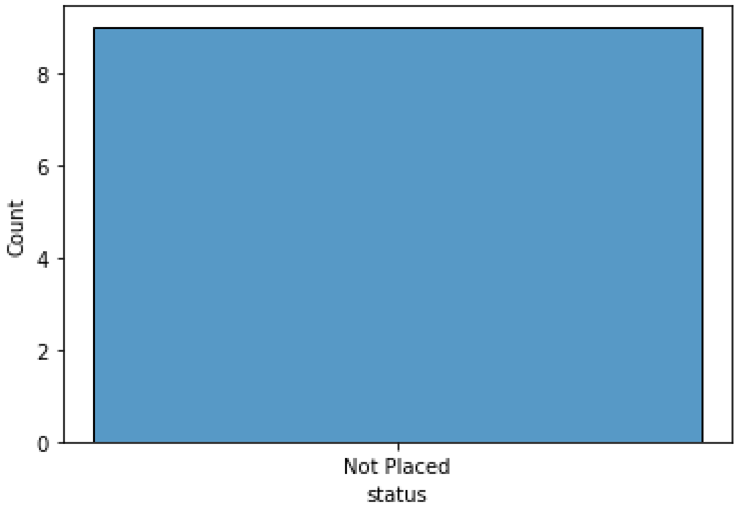
Conclusion :- Eligibility Criteria for scc percentage is above 49% to get placement

In [60]:

```
not_eligible = df[df["ssc_p"] < 49]

sns.histplot(not_eligible["status"])
```

Out[60]:

```
<AxesSubplot:xlabel='status', ylabel='Count'>
```

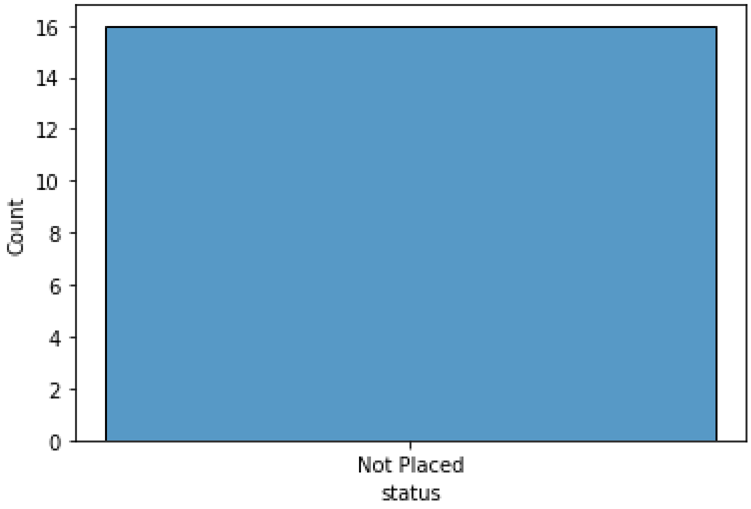


HSC Percentage

Conclusion :- Eligibility Criteria for hsc percentage above 50.83% to get placement

```
In [61]: not_eligible = df[df["hsc_p"] < 50.83]

         sns.histplot(not_eligible["status"])
```

```
Out[61]: <AxesSubplot:xlabel='status', ylabel='Count'>
```

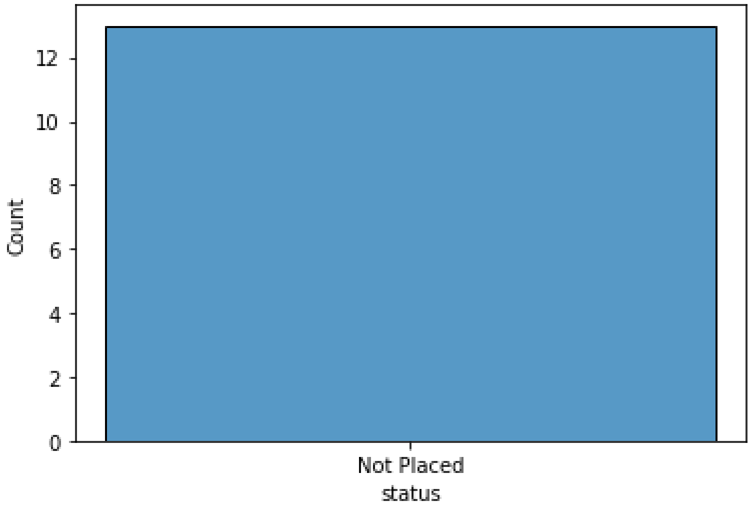


Degree Percentage

Conclusion :- Eligibility Criteria for Degree percentage above 56% to get placement

```
In [62]: not_eligible = df[df["degree_p"] < 56]

         sns.histplot(not_eligible["status"])
```

```
Out[62]: <AxesSubplot:xlabel='status', ylabel='Count'>
```



Etest Percentage

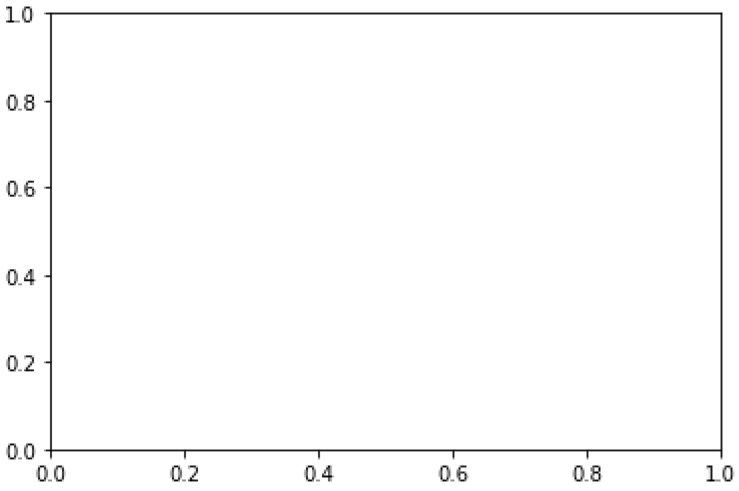
Conclusion :- No Eligibility Criteria for Etest percentage to get placement

In [63]:

```
not_eligible = df[df["etest_p"] < 50]

sns.histplot(not_eligible["status"])
```

Out[63]:

```
<AxesSubplot:>
```



MBA Percentage

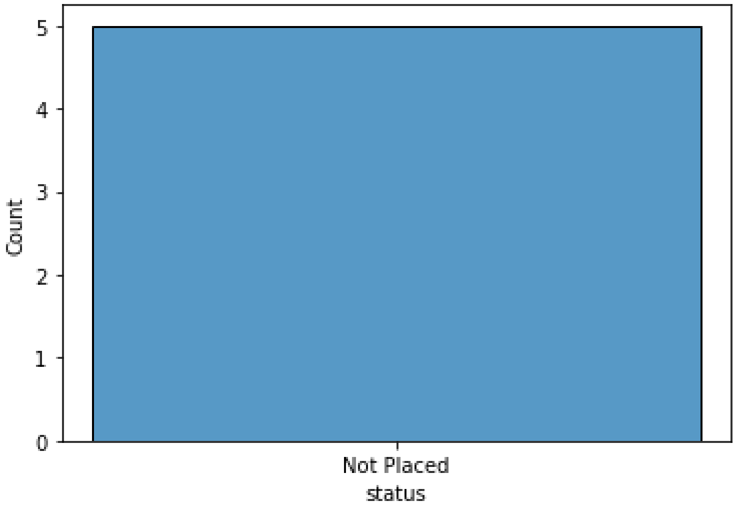
Conclusion :- Eligibility Criteria for MBA percentage above 52.38% to get placement

In [64]:

```
not_eligible = df[df["mba_p"] < 52.38]

sns.histplot(not_eligible["status"])
```

Out[64]:

```
<AxesSubplot:xlabel='status', ylabel='Count'>
```



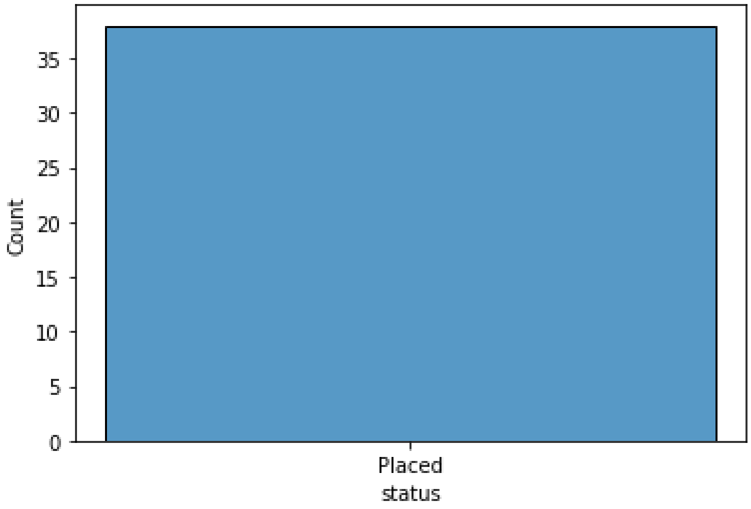
Minimum Merit Score for guarentee placement

SSC Percentage

Conclusion :- Merit Score for SSC percentage is *above 77.8%* to get placement

In [65]:
```
merit = df[df["ssc_p"] > 77.8]

sns.histplot(merit["status"])
```

Out[65]:
```
<AxesSubplot:xlabel='status', ylabel='Count'>
```

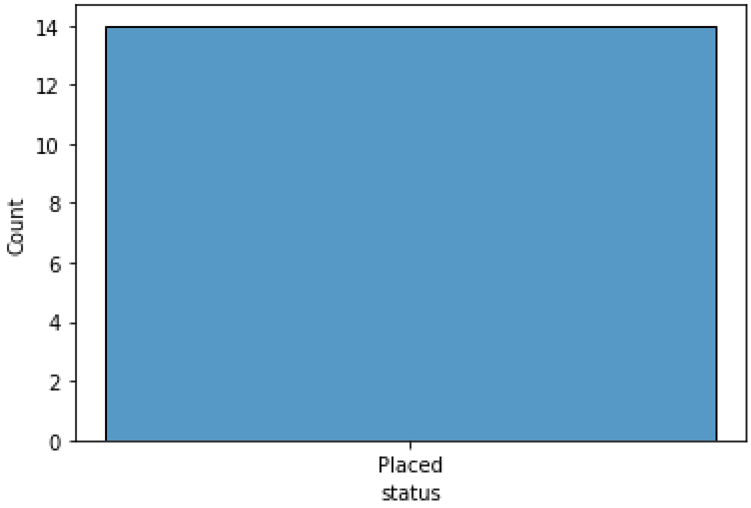


HSC Percentage

Conclusion :- Merit Score for HSC percentage is *above 82%* to get placement

In [66]:
```
merit = df[df["hsc_p"] > 82]

sns.histplot(merit["status"])
```

Out[66]:
```
<AxesSubplot:xlabel='status', ylabel='Count'>
```

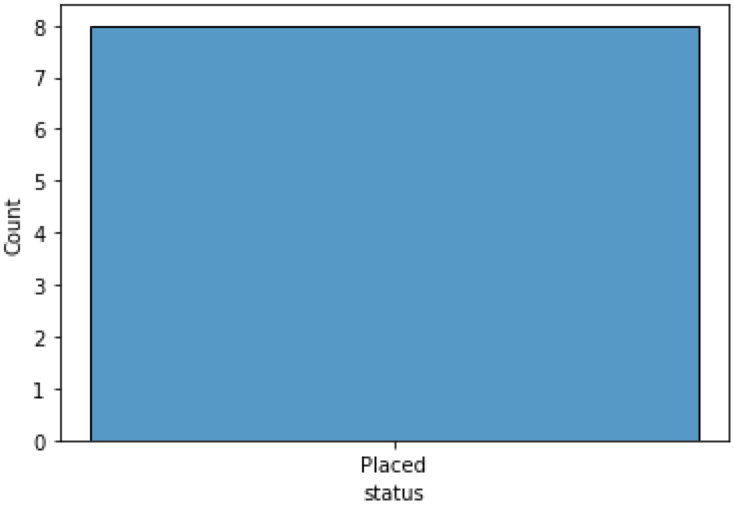


Degree Percentage

Conclusion :- Merit Score for Degree percentage is *above 79%* to get placement

```
In [67]: merit = df[df["degree_p"] > 79]

         sns.histplot(merit["status"])
```

```
Out[67]: <AxesSubplot:xlabel='status', ylabel='Count'>
```

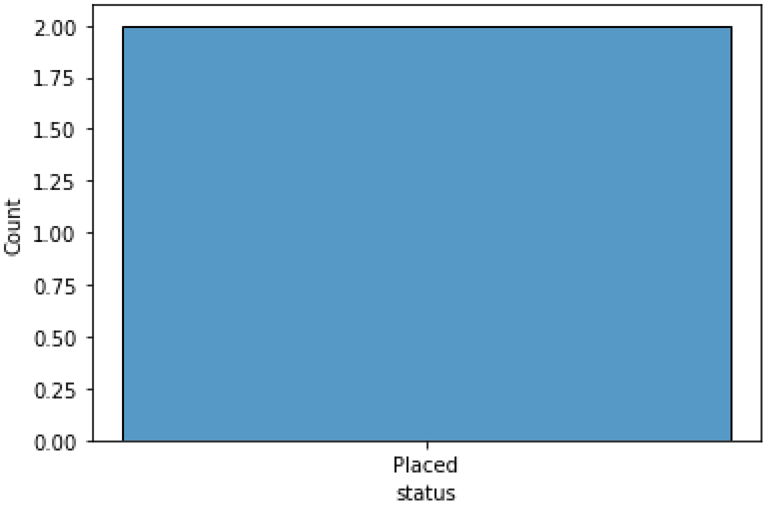


Etest Percentage

Conclusion :- Merit Score for Etest percentage is *above 97%* to get placement

```
In [68]: merit = df[df["etest_p"] > 97]

         sns.histplot(merit["status"])
```

```
Out[68]: <AxesSubplot:xlabel='status', ylabel='Count'>
```

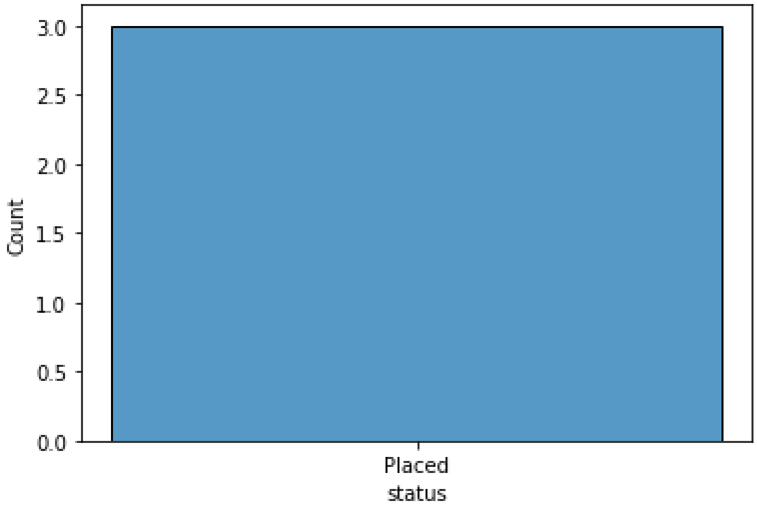


MBA Percentage

Conclusion :- Merit Score for MBA percentage is *above 75.71%* to get placement

```
In [69]: merit = df[df["mba_p"] > 75.71]

         sns.histplot(merit["status"])
```

```
Out[69]: <AxesSubplot:xlabel='status', ylabel='Count'>
```



Q & A

Which factor influenced a candidate in getting placed?

- Candidate with **work experience** have higher chance of getting placed
- Candidate pursuing **hsc subject in Commerce and Science** have chance of getting placed
- Candidate pursuing **degree in Commerce and Management** have higher chance of getting placed
- Candidate pursuing **Specialisation in Market & finance** have higher chance of getting placed

Does percentage matters for one to get placed?

Yes percentage matter for getting placed Here is quick table given below

Feature	Eligibility Criteria	Merit Score Guarentee Placed
ssc_p	Above 49 %	Above 77.8 %
hsc_p	Above 50.83 %	Above 82 %

Feature	Eligibility Criteria	Merit Score Guarentee Placed
degree_p	Above 56 %	Above 79 %
etest_p	Not Criteria	Above 97 %

Which degree specialization in much demanded by corporate?

1. Commerce and Management
2. Market & Finance