# MINOR PROJECT
# READING THE DATASET

```
#TASK 1:EXPLORATORY DATA ANANLYSIS


import pandas as pd
mpdf=pd.read_csv(r"C:\Users\anany\Downloads\minor project.csv")
mpdf.head()
```

In [4]:

```
#CHECKING THE DATASET FOR ANY NULL OR MISSING VALUES
mpdf.isna().sum()
```

Out[4]:

```
sl_no              0
gender             0
ssc_p              0
ssc_b              0
hsc_p              0
hsc_b              0
hsc_s              0
degree_p           0
degree_t           0
workex             0
etest_p            0
specialisation     0
mba_p              0
status             0
salary            67
dtype: int64
```

In [6]:

```
#SINCE THE SALARY HAD MANY MISSING VALUES FOR NON PLACED STUDENTS WE REPLACED IT BY 0 FOR THOSE STUDI
mpdf['salary'] = mpdf['salary'].fillna(0)
mpdf['salary'].isnull().sum()
```

Out[6]:

```
0
```

In [7]:

```
mpdf.isna().sum()
```

Out[7]:

```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

In [8]:

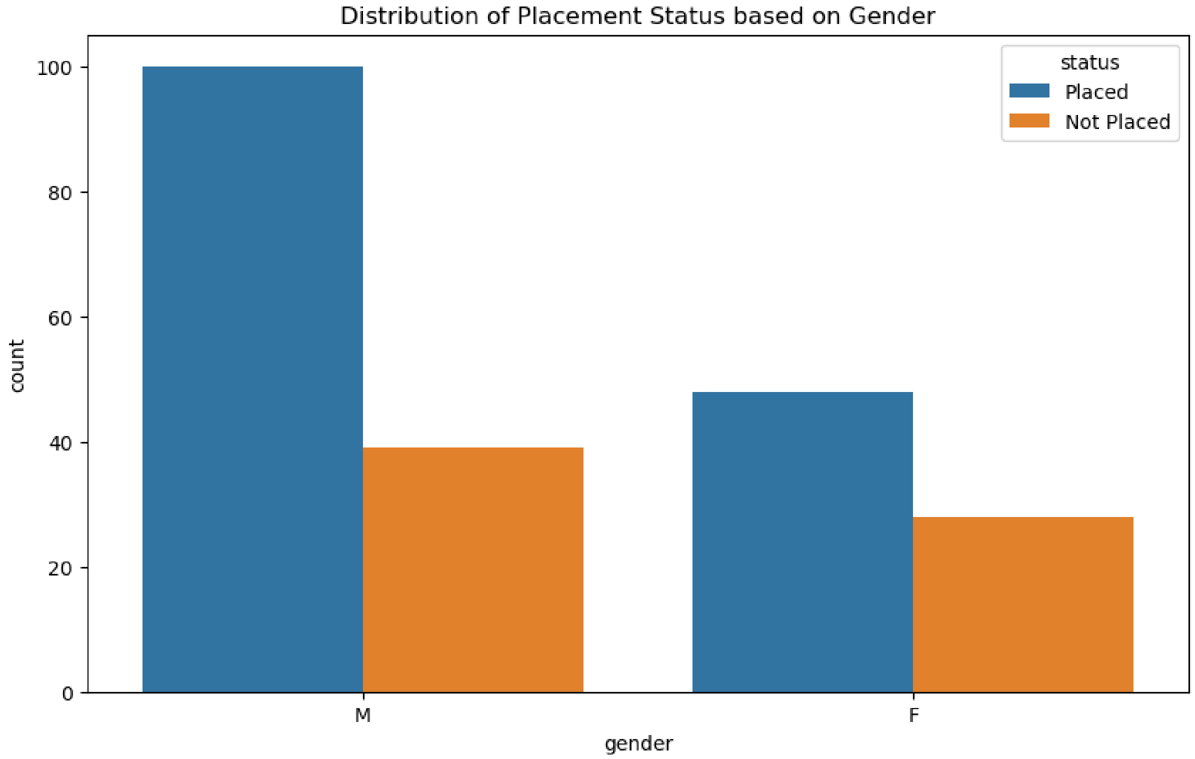
```
mpdf.head()
```

Out[8]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	spec
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	

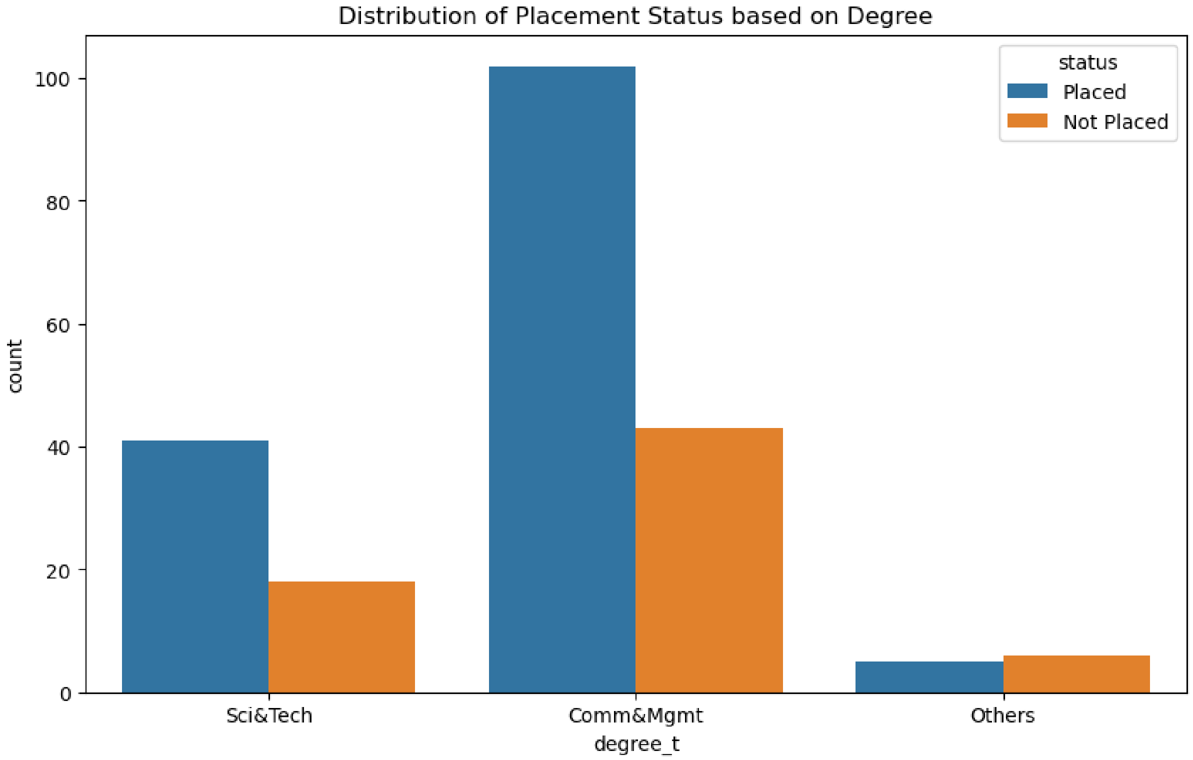
In [10]:

```
#DISTRIBUTION OF VARIOUS CATEGORICAL VARIABLES
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.countplot(x='gender', hue='status', data=mpdf)
plt.title('Distribution of Placement Status based on Gender')
plt.show()

#ACCORDING TO THE GRAPH FEMALES PLACED ARE RELATIVELY LOW
```



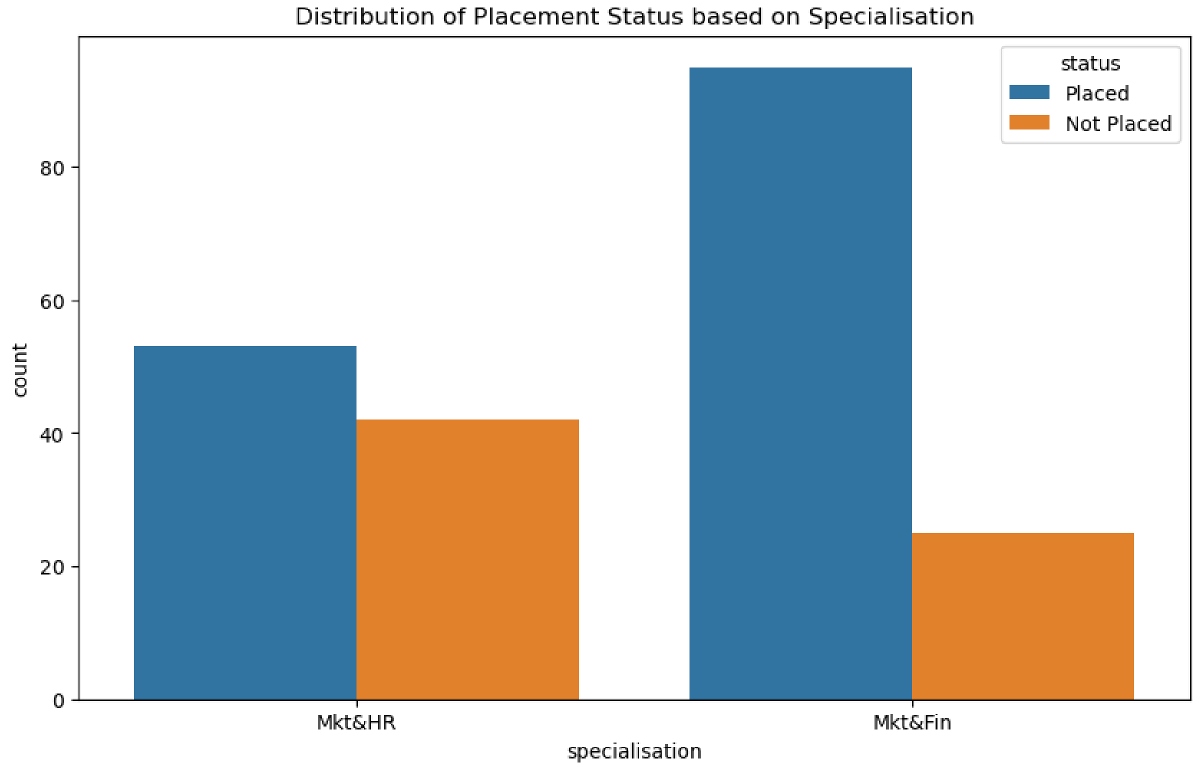
In [12]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='degree_t', hue='status', data=mpdf)
plt.title('Distribution of Placement Status based on Degree')
plt.show()

#COMMERCE AND MANAGEMENT HAS HIGHEST STUDENTS PLACED
```



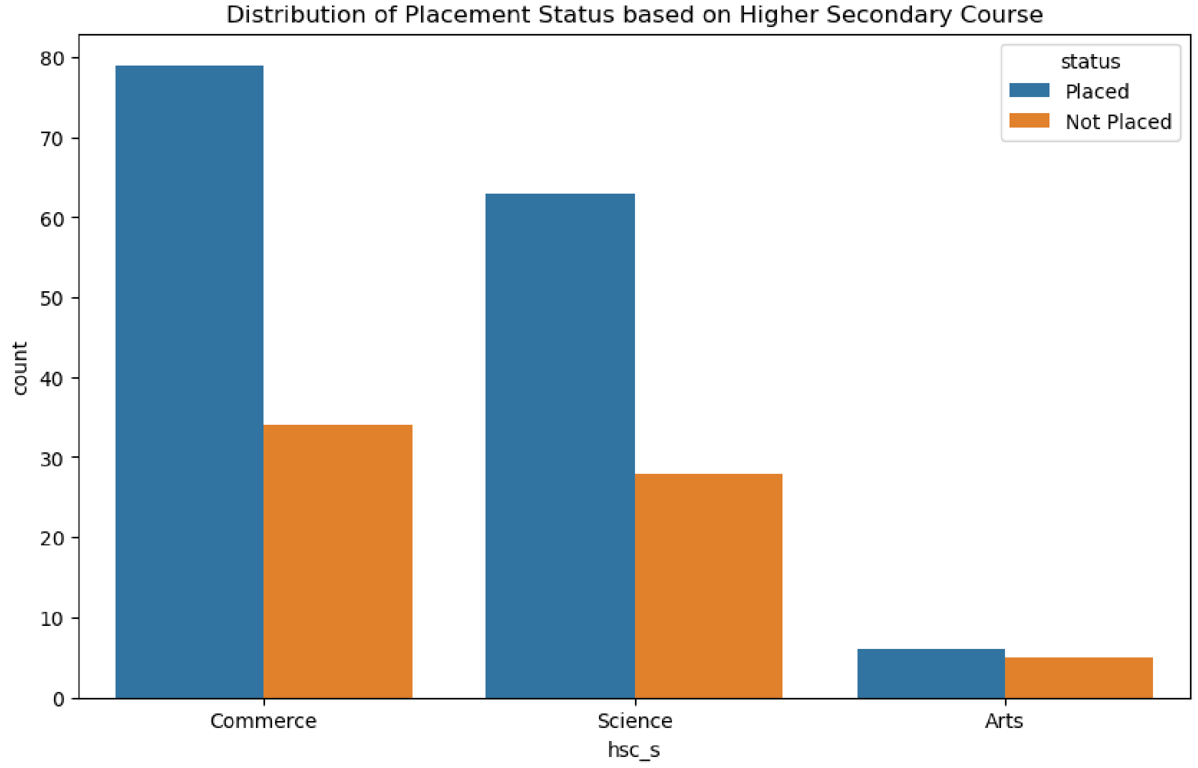
In [11]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='specialisation', hue='status', data=mpdf)
plt.title('Distribution of Placement Status based on Specialisation')
plt.show()
```



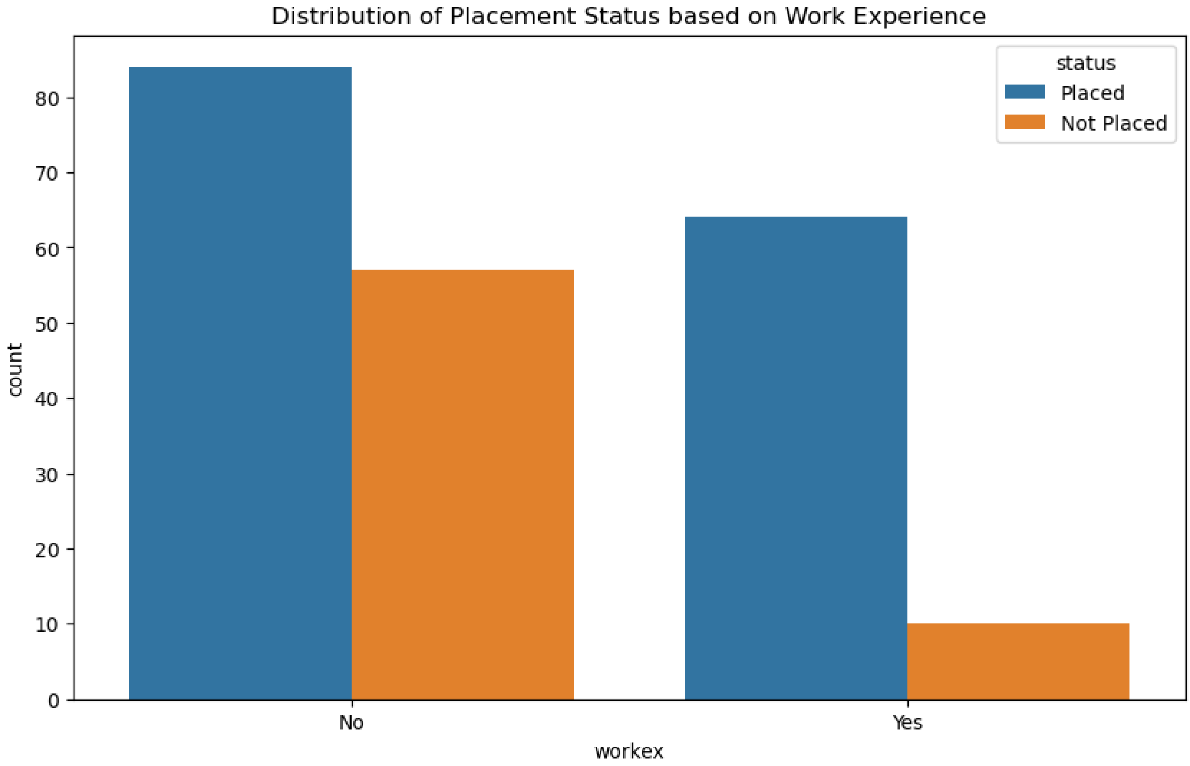
In [13]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='hsc_s', hue='status', data=mpdf)
plt.title('Distribution of Placement Status based on Higher Secondary Course')
plt.show()
```



In [14]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='workex', hue='status', data=mpdf)
plt.title('Distribution of Placement Status based on Work Experience')
plt.show()

#IT HAS BEEN SENN THAT HIGHER NUMBER OF PEOPLE ARE GETTING PLACED WHO HAVE NO WORK EXPERIENCE.
```



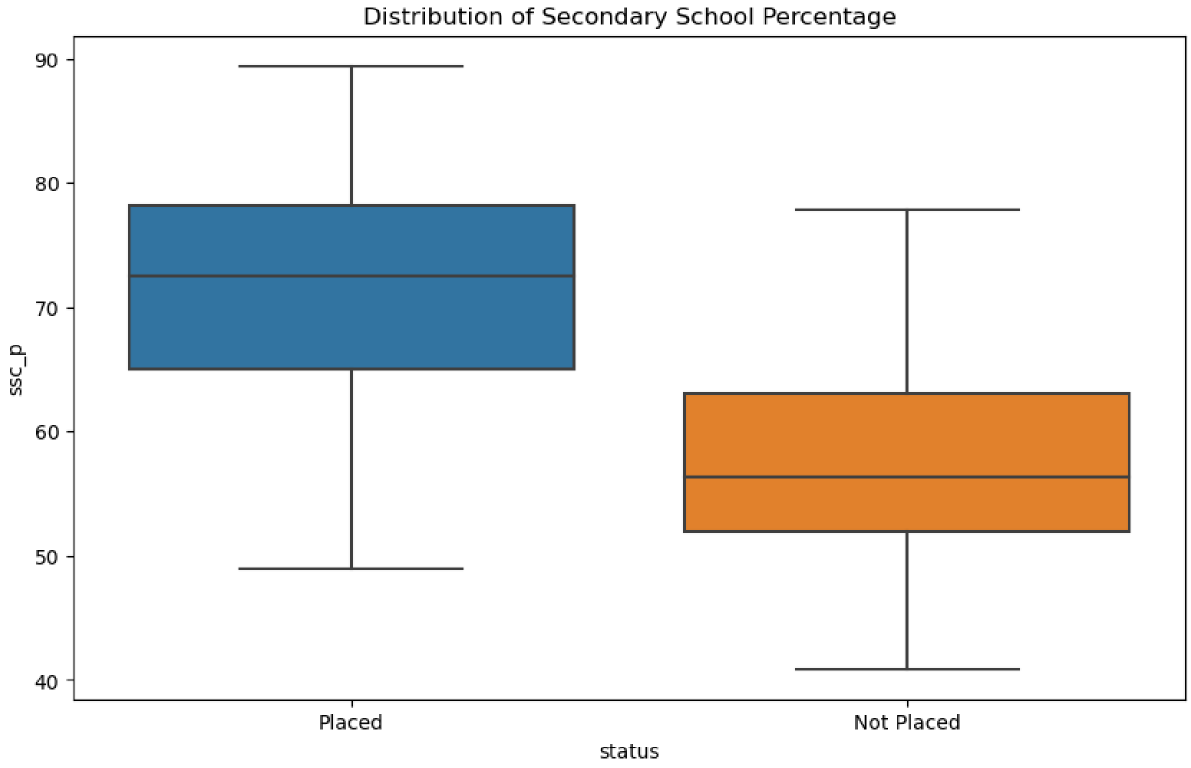
In [15]:

```
correlation = mpdf[['ssc_p', 'hsc_p', 'degree_p', 'workex', 'etest_p', 'mba_p', 'status']].corr()
print("Correlation Matrix:\n", correlation)
```

```
Correlation Matrix:
               ssc_p     hsc_p  degree_p   etest_p     mba_p
ssc_p     1.000000  0.511472  0.538404  0.261993  0.388478
hsc_p     0.511472  1.000000  0.434206  0.245113  0.354823
degree_p  0.538404  0.434206  1.000000  0.224470  0.402364
etest_p   0.261993  0.245113  0.224470  1.000000  0.218055
mba_p     0.388478  0.354823  0.402364  0.218055  1.000000
```

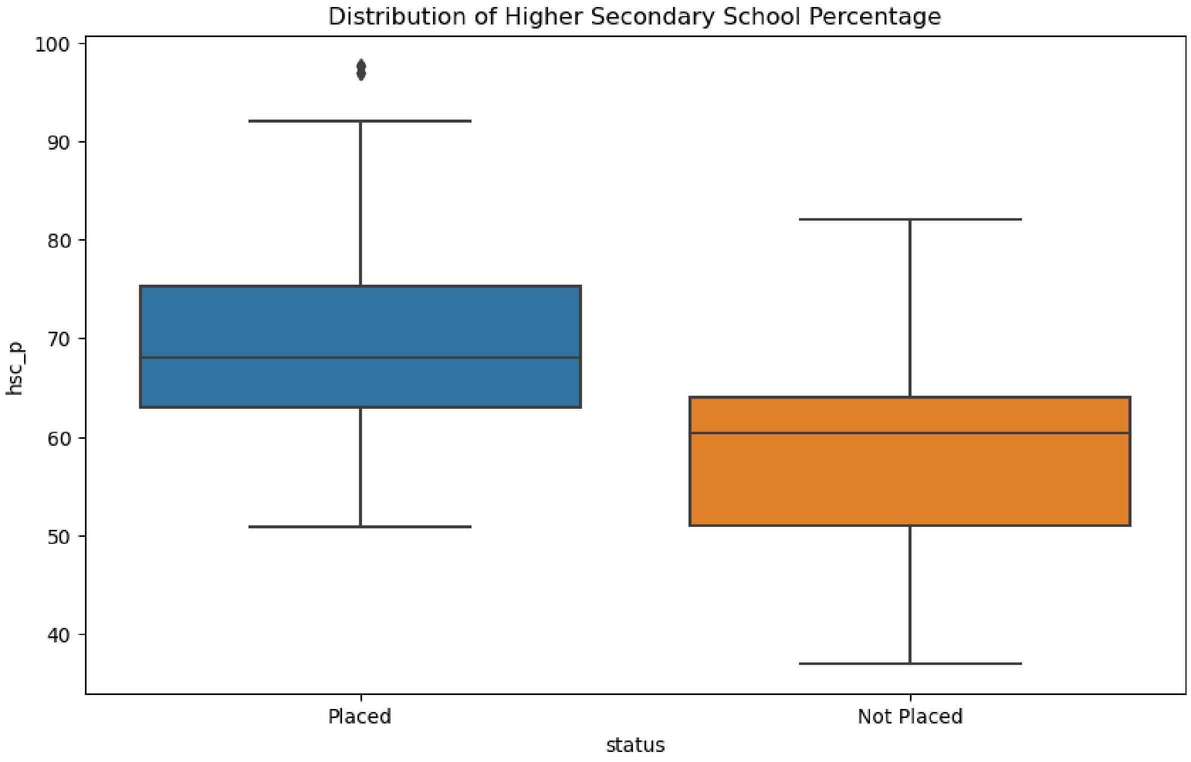
In [16]:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='status', y='ssc_p', data=mpdf)
plt.title('Distribution of Secondary School Percentage')
plt.show()
```



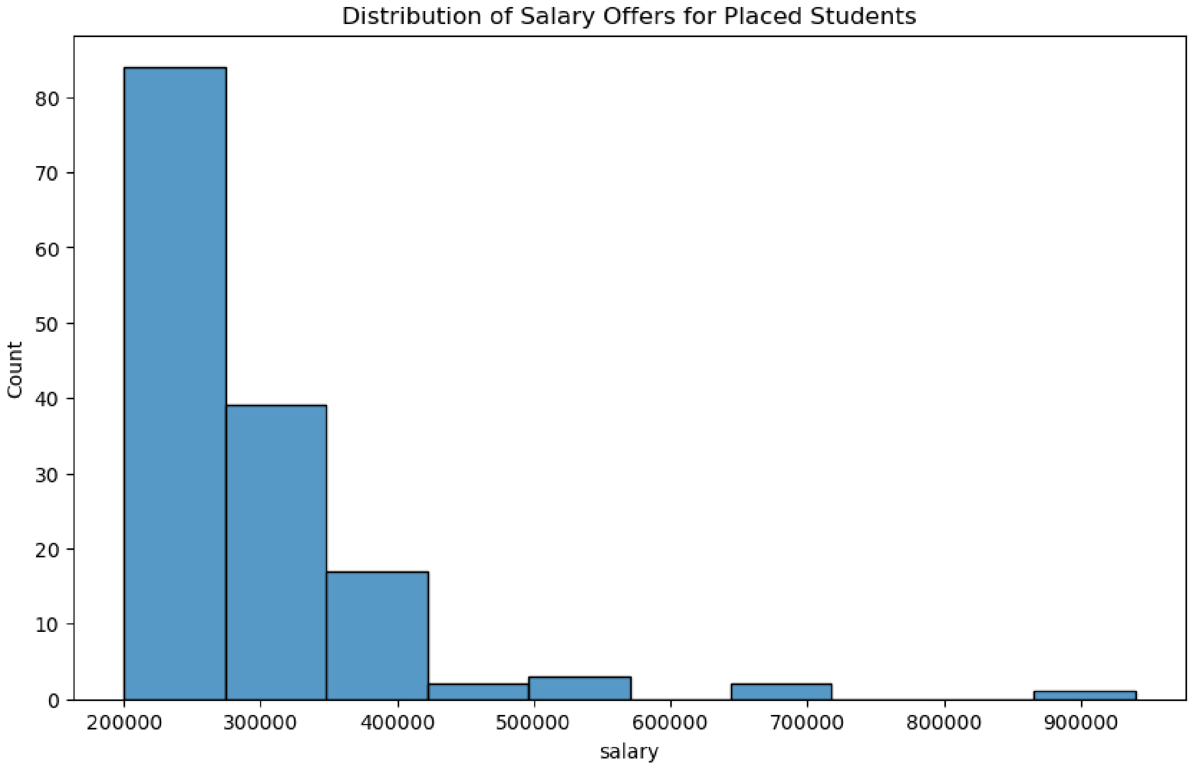
In [17]:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='status', y='hsc_p', data=mpdf)
plt.title('Distribution of Higher Secondary School Percentage')
plt.show()
```



In [18]:

```
#distribution of the salary offers for placed students
placed_data = mpdf[mpdf['status'] == 'Placed']
plt.figure(figsize=(10, 6))
sns.histplot(placed_data['salary'], bins=10)
plt.title('Distribution of Salary Offers for Placed Students')
plt.show()
```



In [19]:

```
# minimum, maximum, mean, and median salary values
salary_stats = placed_data['salary'].describe()
print("Salary Statistics:\n", salary_stats)
```

```
Salary Statistics:
 count      148.000000
mean     288655.405405
std       93457.452420
min      200000.000000
25%      240000.000000
50%      265000.000000
75%      300000.000000
max      940000.000000
Name: salary, dtype: float64
```

# DATA PREPROCESSING AND NORMALIZATION

```
features = mpdf.drop('status', axis=1)
target = mpdf['status']
```

In [23]:

```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_19500\1479815098.py in <module>
----> 1 X_train, X_test, y_train, y_test = train_test_split(features, target, test_siz
e=0.2, random_state=42)

NameError: name 'train_test_split' is not defined
```

In [24]:

```
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42
```

In [ ]:

```

```