In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # Data Visualization
import seaborn as sns # Data Visualization
```

In [2]:

```
df = pd.read_csv('collegePlace.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

In [4]:

```
df.shape
```

Out[4]:

```
(2966, 8)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Age                2966 non-null   int64 
 1   Gender             2966 non-null   object
 2   Stream             2966 non-null   object
 3   Internships        2966 non-null   int64 
 4   CGPA               2966 non-null   int64 
 5   Hostel             2966 non-null   int64 
 6   HistoryOfBacklogs  2966 non-null   int64 
 7   PlacedOrNot        2966 non-null   int64 
dtypes: int64(6), object(2)
memory usage: 185.5+ KB
```

In [6]:

```
df.isna().sum()
```

Out[6]:

```
Age                  0
Gender               0
Stream               0
Internships          0
CGPA                 0
Hostel               0
HistoryOfBacklogs    0
PlacedOrNot          0
dtype: int64
```

In [7]:

```
df.Stream.unique()
```

Out[7]:

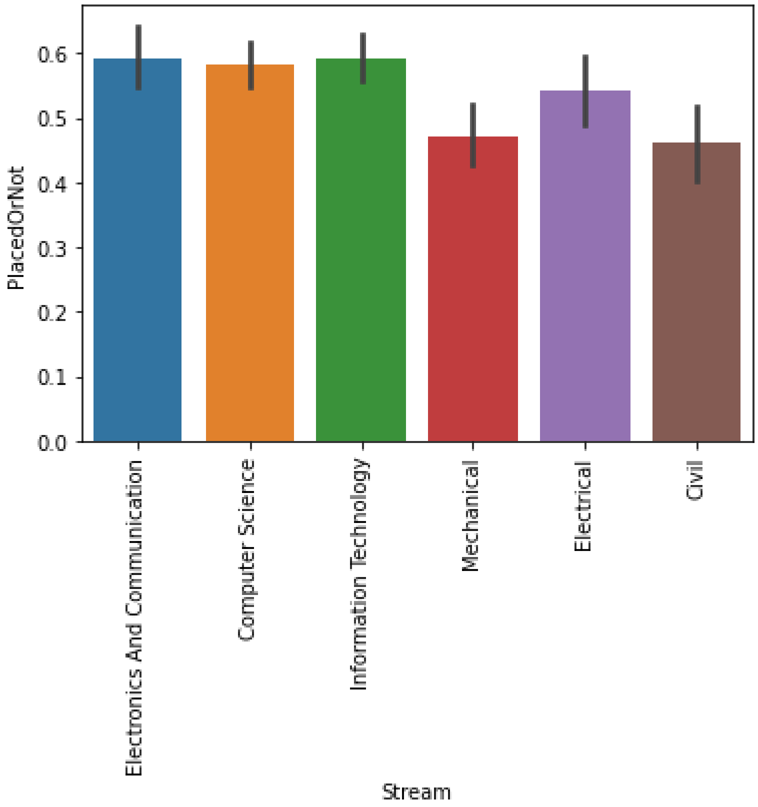
```
array(['Electronics And Communication', 'Computer Science',
       'Information Technology', 'Mechanical', 'Electrical', 'Civil'],
      dtype=object)
```

In [8]:

```
plt.xticks(rotation = 90)
sns.barplot(x = df.Stream, y = df.PlacedOrNot)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1190c5cd0>
```



In [9]:

```
df.Age.unique()
```

Out[9]:

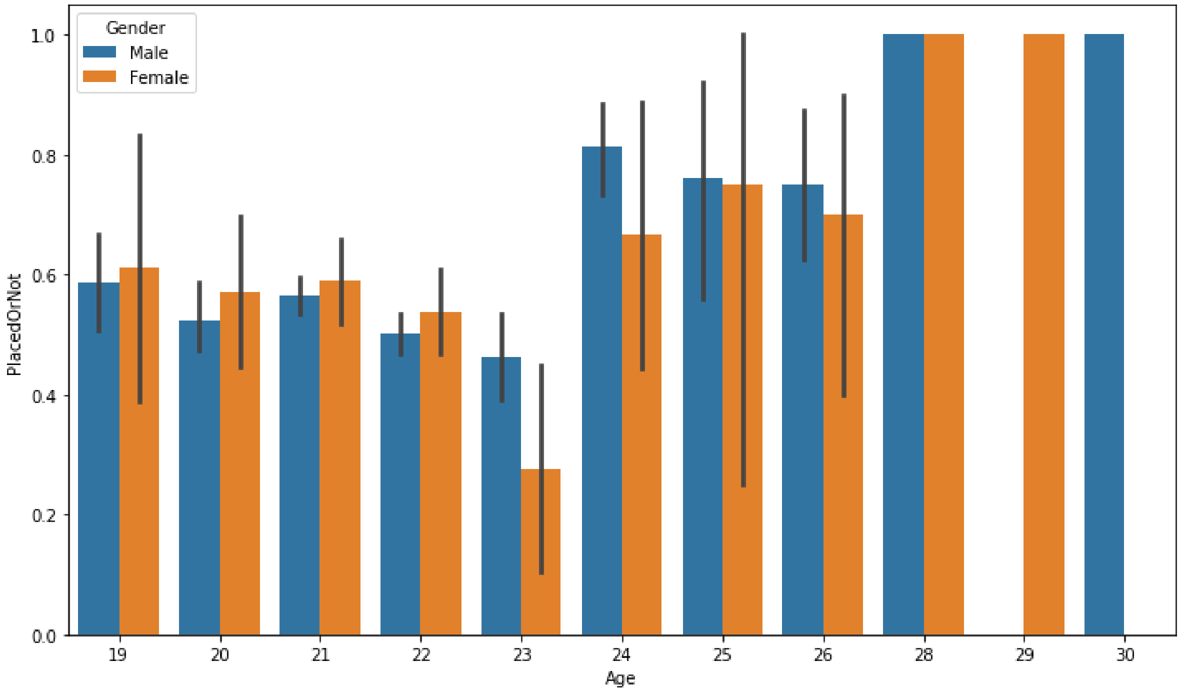
```
array([22, 21, 23, 24, 28, 30, 25, 26, 20, 19, 29], dtype=int64)
```

In [10]:

```
plt.figure(figsize = (12,7))
sns.barplot(x = df.Age, y = df.PlacedOrNot, hue = df.Gender)
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x104d07c10>
```

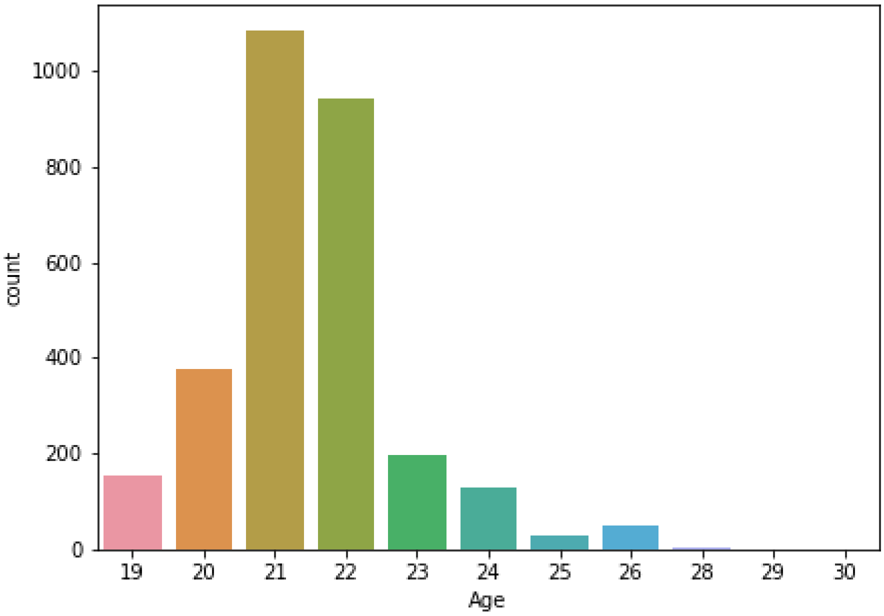


In [11]:

```
plt.figure(figsize = (7,5))
sns.countplot(x = df.Age)
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x104cca4f0>
```



In [12]:

```
df.Age.value_counts()
```

Out[12]:

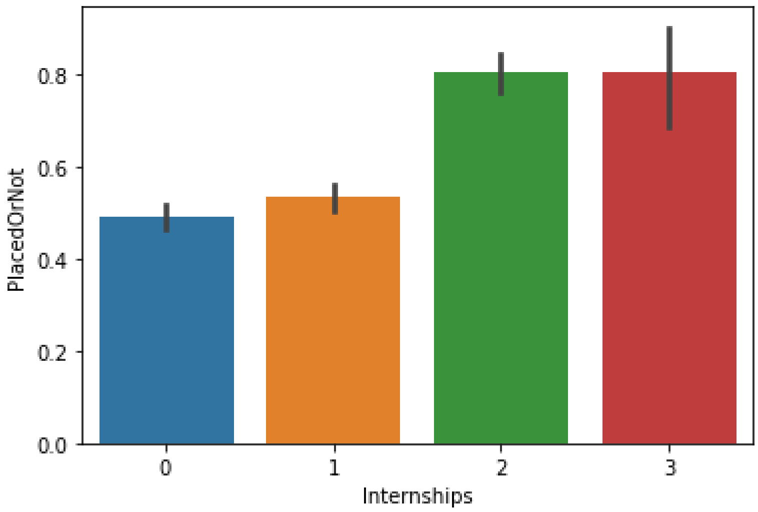
```
21    1084
22     941
20     375
23     195
19     156
24     131
26      50
25      29
28       3
29       1
30       1
Name: Age, dtype: int64
```

In [13]:

```
sns.barplot(x = df.Internships, y = df.PlacedOrNot)
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1191597f0>
```



In [14]:

```
df.Internships.value_counts()
```

Out[14]:

```
0    1331
1    1234
2     350
3      51
Name: Internships, dtype: int64
```

In [15]:

```
df.CGPA.value_counts()
```

Out[15]:

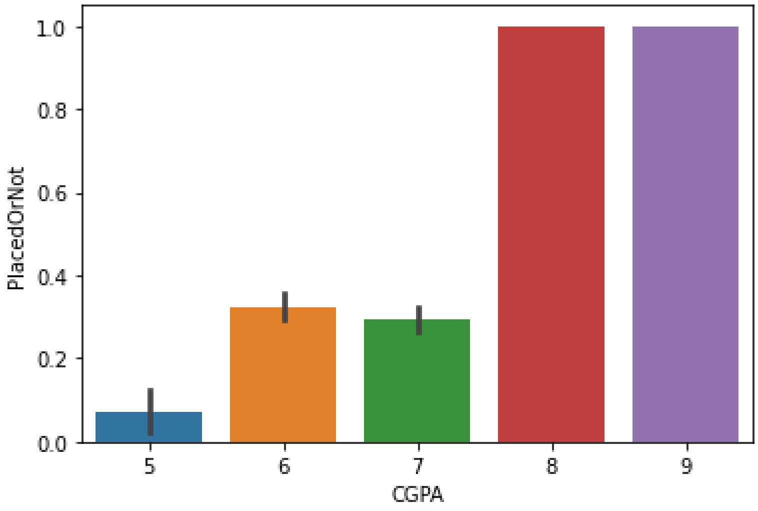
```
7    956
8    915
6    834
9    165
5     96
Name: CGPA, dtype: int64
```

In [16]:

```
sns.barplot(x = df.CGPA, y = df.PlacedOrNot)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1191c4490>
```

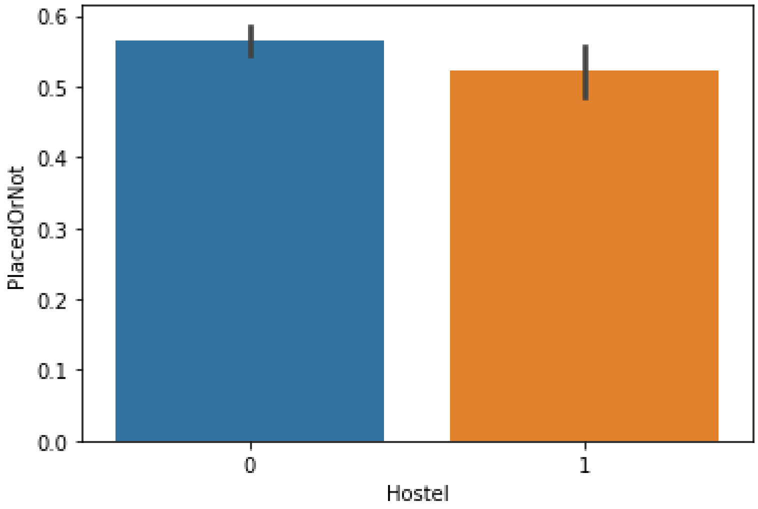


In [17]:

```
sns.barplot(x = df.Hostel, y = df.PlacedOrNot)
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x119224b20>
```

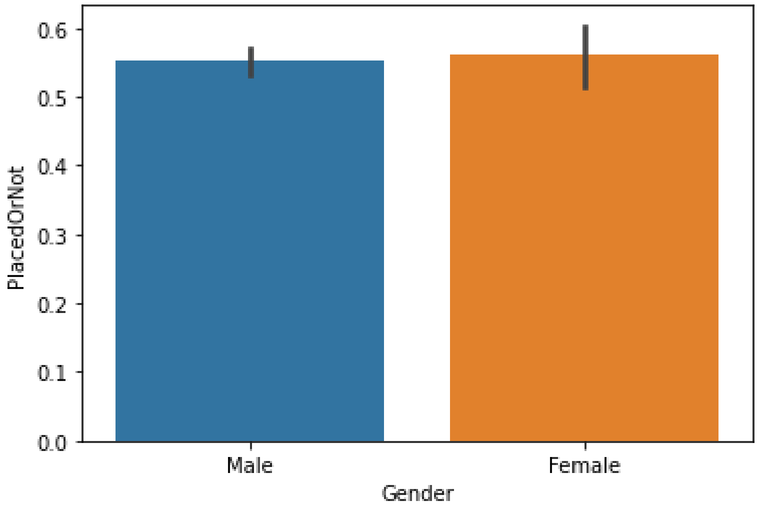


In [18]:

```
sns.barplot(x = df.Gender, y = df.PlacedOrNot)
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11927ffd0>
```

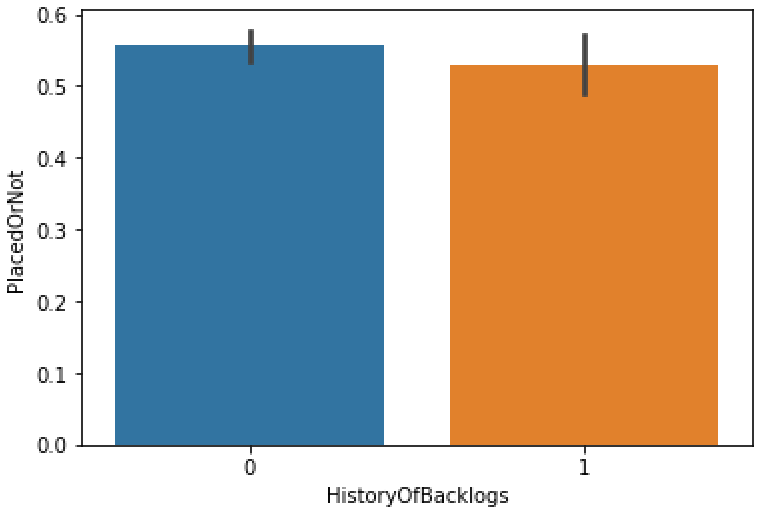


In [19]:

```
sns.barplot(x = df.HistoryOfBacklogs, y = df.PlacedOrNot)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1192d6250>
```



In [20]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [21]:

```
df.Gender = le.fit_transform(df.Gender)
df.Stream = le.fit_transform(df.Stream)
```

In [22]:

```
df.head()
```

Out[22]:

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	1	3	1	8	1	1	1
1	21	0	1	0	7	1	1	1
2	22	0	4	1	6	0	0	1
3	21	1	4	0	8	0	1	1
4	22	1	5	0	8	1	0	1

In [23]:

```
x = df.drop(['PlacedOrNot'], axis = 1)
```

In [24]:

```
y = df.PlacedOrNot
```

In [25]:

```
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

In [26]:

```
from sklearn.model_selection import cross_val_score
```

In [27]:

```
cross_val_score(SVC(),x, y, cv = 3)
```

Out[27]:

```
array([0.73609707, 0.76238625, 0.84817814])
```

In [28]:

```
cross_val_score(DecisionTreeClassifier(), x, y, cv = 3)
```

Out[28]:

```
array([0.84428716, 0.83923155, 0.90890688])
```

In [29]:

```
cross_val_score(LogisticRegression(), x, y, cv = 3)
```

Out[29]:

```
array([0.71991911, 0.74823054, 0.83704453])
```

In [30]:

```
cross_val_score(RandomForestClassifier(n_estimators=50), x, y, cv = 3)
```

Out[30]:

```
array([0.8463094 , 0.85338726, 0.89473684])
```

In [31]:

```
cross_val_score(KNeighborsClassifier(),x, y ,cv = 3)
```

Out[31]:

```
array([0.82912032, 0.81193124, 0.88259109])
```

In [32]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [33]:

```
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Out[33]:

```
RandomForestClassifier()
```

In [34]:

```
y_pred = model.predict(X_test)
```

In [35]:

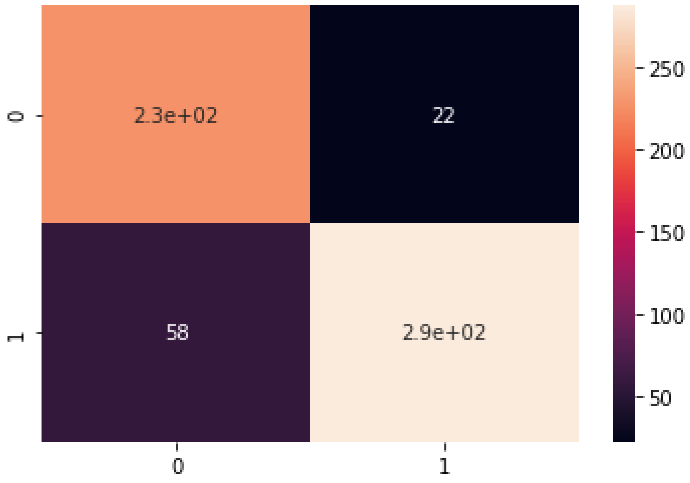
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [36]:

```
sns.heatmap(cm, annot = True)
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11a05f3d0>
```



In [37]:

```
print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.9283305227655987
Testing Accuracy : 0.8653198653198653
```