In [53]:
```
import pandas as pd
```

In [54]:
```
df = pd.read_csv(r'C:\Users\Nitika\Downloads\placement.csv')
```

In [3]:
```
df
```

Out[3]:

	cgpa	placement_exam_marks	placed
0	7.19	26	yes
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

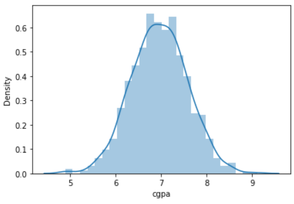
1000 rows × 3 columns

In [4]:
```
import seaborn as sns
```

In [5]:
```
sns.distplot(df['cgpa'])
```

```
C:\Users\Nitika\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

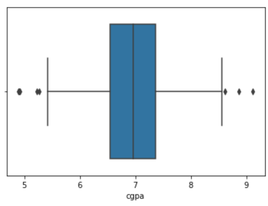
Out[5]:
```
<AxesSubplot:xlabel='cgpa', ylabel='Density'>
```



In [6]:
```
sns.boxplot(df['cgpa'])
```

```
C:\Users\Nitika\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[6]:
```
<AxesSubplot:xlabel='cgpa'>
```



In [9]:
```
df["cgpa"].max()
```

Out[9]:
```
9.12
```

In [10]:
```
df['cgpa'].min()
```

Out[10]:
```
4.89
```

In [11]:
```
df['cgpa'].std()
```

Out[11]:
```
0.6158978751323894
```

In [12]:
```
df['cgpa'].mean()
```

Out[12]:
```
6.96124000000001
```

In [14]:
```
lowerlimit = df['cgpa'].mean() - 3 * df['cgpa'].std()
```

In [15]:
```
lowerlimit
```

Out[15]:
```
5.113546374602842
```

In [17]:
```
df[df['cgpa']<5.11]
```

Out[17]:

	cgpa	placement_exam_marks	placed
485	4.92	44	1
997	4.89	34	0
999	4.90	10	1

In [19]:
```
upperlimit = df['cgpa'].mean() + 3 * df['cgpa'].std()
```

In [38]:
```
upperlimit
```

Out[38]:
```
8.808933625397177
```

In [39]:
```
df[df['cgpa']> 8.8]
```

Out[39]:

	cgpa	placement_exam_marks	placed
995	8.87	44	1
996	9.12	65	1

In [40]:
```
df[(df['cgpa']>8.8)| (df['cgpa']< 5)]
```

Out[40]:

	cgpa	placement_exam_marks	placed
485	4.92	44	1
995	8.87	44	1
996	9.12	65	1
997	4.89	34	0
999	4.90	10	1

Trimming & Capping

Trimming = delete the outliers & Capping = Use the outlier in same data

df[(df['cgpa']<8.8) & (df['cgpa']>5)]

In [55]:
```
newdf = df[(df['cgpa']<8.8) & (df['cgpa']>5)]
```

In [56]:
```
newdf
```

Out[56]:

	cgpa	placement_exam_marks	placed
0	7.19	26	yes
1	7.46	38	1
2	7.54	40	1
3	6.42	8	1
4	7.23	17	0
...	...	...	...
991	7.04	57	0
992	6.26	12	0
993	6.73	21	1
994	6.48	63	0
998	8.62	46	1

995 rows × 3 columns

In [57]:
```
newdf.shape
```

Out[57]:
```
(995, 3)
```

capping

In [58]:
```
upperlimit
```

Out[58]:
```
8.808933625397177
```

In [59]:
```
lowerlimit
```

Out[59]:
```
5.113546374602842
```

In [60]:
```

```