In [2]:

```
import pandas as pd
df=pd.read_csv("walmart.csv")
```

In [3]:

```
df
```

Out[3]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marit
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	
...	...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

550068 rows × 10 columns

In [4]:

```
df.shape
```

Out[4]:

```
(550068, 10)
```

In [6]:

```
#Trx done by diff Gen
df.groupby('Gender')['Purchase'].describe()
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Gender								
F	135809.0	8734.565765	4767.233289	12.0	5433.0	7914.0	11400.0	23959.0
M	414259.0	9437.526040	5092.186210	12.0	5863.0	8098.0	12454.0	23961.0

In [9]:

```
df.groupby('Gender')['User_ID'].nunique()
```

Out[9]:

```
Gender
F     1666
M     4225
Name: User_ID, dtype: int64
```

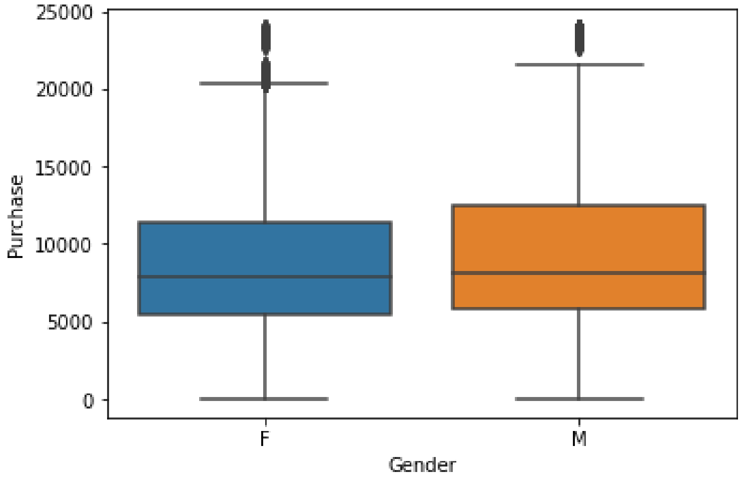
In [10]:

```
import seaborn as sbn


sbn.boxplot(x='Gender', y='Purchase', data=df)
```

Out[10]:

```
<AxesSubplot:xlabel='Gender', ylabel='Purchase'>
```

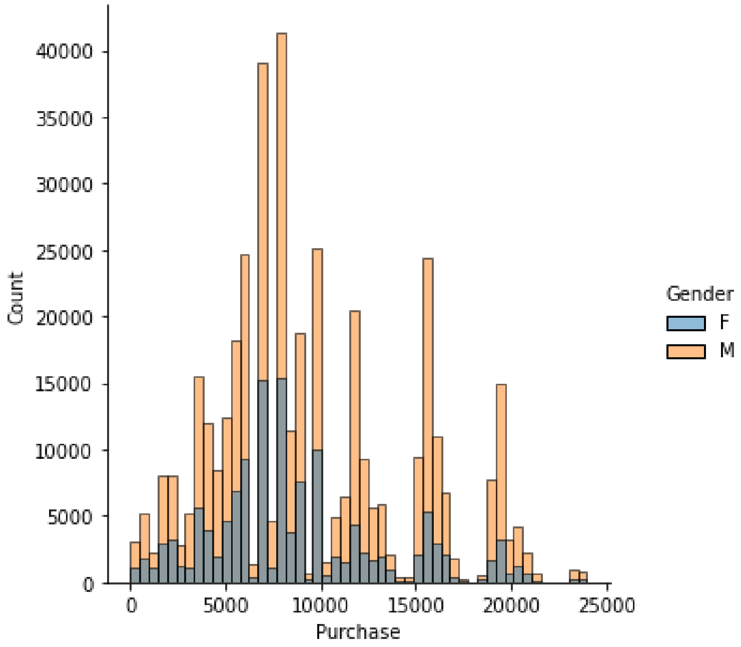


In [15]:

```
sbn.displot(x='Purchase', data=df, bins=50, hue='Gender')
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff56a2cd4c0>
```



In [16]:

```
df.groupby('Gender')['Purchase'].describe()
```

Out[16]:

	count	mean	std	min	25%	50%	75%	max
Gender								
F	135809.0	8734.565765	4767.233289	12.0	5433.0	7914.0	11400.0	23959.0
M	414259.0	9437.526040	5092.186210	12.0	5863.0	8098.0	12454.0	23961.0

In [21]:

```
sample_size=300
iterations=1000
```

In [22]:

```

male_sample_means=[df[df['Gender']=='M'].sample(sample_size)['Purchase'].mean() for i in 
```

In [24]:

```
import numpy as np

np.mean(male_sample_means)
```

Out[24]:

```
9441.886329999998
```

In [25]:

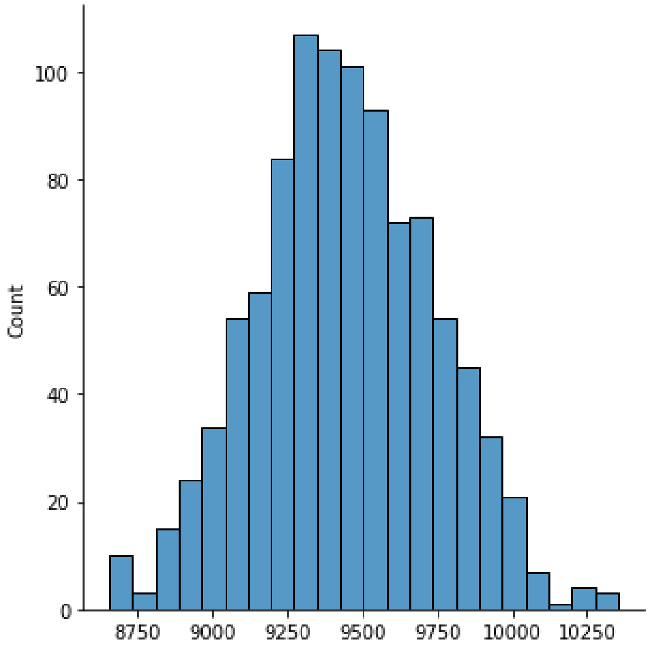
```
female_sample_means=[df[df['Gender']=='F'].sample(sample_size)['Purchase'].mean() for i i
```

In [26]:

```
sbn.displot(male_sample_means)
```

Out[26]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff56b360b20>
```

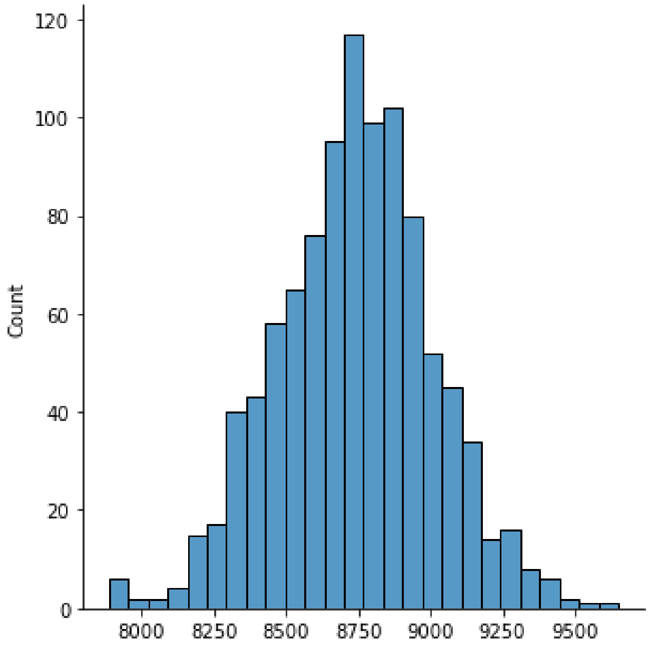


In [27]:

```
sbn.displot(female_sample_means)
```

Out[27]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff56b5ccfd0>
```



In [29]:

```
lower_male_sample_means= np.mean(male_sample_means) - 1.96*np.std(male_sample_means)
```

In [30]:

```
upper_male_sample_means= np.mean(male_sample_means) + 1.96*np.std(male_sample_means)
```

In [31]:

```
lower_female_sample_means= np.mean(female_sample_means) - 1.96*np.std(female_sample_means
```

In [32]:

```
upper_female_sample_means= np.mean(female_sample_means) + 1.96*np.std(female_sample_means
```

In [33]:

```
#ci for males

lower_male_sample_means, upper_male_sample_means
```

Out[33]:

```
(8866.991517959994, 10016.781142040001)
```

In [34]:

```
lower_female_sample_means, upper_female_sample_means
```

Out[34]:

```
(8211.76883295025, 9266.111780383084)
```

In [36]:

```
np.percentile(male_sample_means,[2.5,97.5])
```

Out[36]:

```
array([ 8879.84708333, 10005.78366667])
```

In [ ]:

```
#remove the overlap

1. increase the sample size
2. 90% confidence, 99% confidence
```