In [31]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

In [32]:

```python
path = ("/Users/sudhanshubiswal/Downloads/Salary_Dataset_with_Extra_Features.csv
df = pd.read_csv(path)
```

In [33]:

```python
df.head(10)
```

Out[33]:

	Rating	Company Name	Job Title	Salary	Salaries Reported	Location	Employment Status	Job Roles
0	3.8	Sasken	Android Developer	400000	3	Bangalore	Full Time	Android
1	4.5	Advanced Millennium Technologies	Android Developer	400000	3	Bangalore	Full Time	Android
2	4.0	Unacademy	Android Developer	1000000	3	Bangalore	Full Time	Android
3	3.8	SnapBizz Cloudtech	Android Developer	300000	3	Bangalore	Full Time	Android
4	4.4	Appoids Tech Solutions	Android Developer	600000	3	Bangalore	Full Time	Android
5	4.2	Freelancer	Android Developer	100000	3	Bangalore	Full Time	Android
6	3.7	SQUARE N CUBE	Android Developer	192000	3	Bangalore	Full Time	Android
7	3.1	Samsung R&D Institute India - Bangalore	Android Developer	400000	3	Bangalore	Full Time	Android
8	3.7	DXMinds Technologies	Android Developer	300000	3	Bangalore	Full Time	Android
9	3.6	Endeavour Software Technologies	Android Developer	600000	3	Bangalore	Full Time	Android

In [34]:

```
df.tail()
```

Out[34]:

	Rating	Company Name	Job Title	Salary	Salaries Reported	Location	Employment Status	Job Roles
22765	4.7	Expert Solutions	Web Developer	200000	1	Bangalore	Full Time	Web
22766	4.0	Nextgen Innovation Labs	Web Developer	300000	1	Bangalore	Full Time	Web
22767	4.1	Fresher	Full Stack Web Developer	192000	13	Bangalore	Full Time	Web
22768	4.1	Accenture	Full Stack Web Developer	300000	7	Bangalore	Full Time	Web
22769	3.8	Thomson Reuters	Associate Web Developer	300000	7	Bangalore	Full Time	Web

In [35]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22770 entries, 0 to 22769
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Rating             22770 non-null  float64
 1   Company Name       22770 non-null  object 
 2   Job Title          22770 non-null  object 
 3   Salary             22770 non-null  int64  
 4   Salaries Reported  22770 non-null  int64  
 5   Location           22770 non-null  object 
 6   Employment Status  22770 non-null  object 
 7   Job Roles          22770 non-null  object 
dtypes: float64(1), int64(2), object(5)
memory usage: 1.4+ MB
```

In [36]:

```
df.shape
```

Out[36]:

```
(22770, 8)
```

In [37]:

```
df.describe()
```

Out[37]:

	Rating	Salary	Salaries Reported
count	22770.000000	2.277000e+04	22770.000000
mean	3.918213	6.953872e+05	1.855775
std	0.519675	8.843990e+05	6.823668
min	1.000000	2.112000e+03	1.000000
25%	3.700000	3.000000e+05	1.000000
50%	3.900000	5.000000e+05	1.000000
75%	4.200000	9.000000e+05	1.000000
max	5.000000	9.000000e+07	361.000000

In [38]:

```
df.columns
```

Out[38]:

```
Index(['Rating', 'Company Name', 'Job Title', 'Salary', 'Salaries Repo
rted',
       'Location', 'Employment Status', 'Job Roles'],
      dtype='object')
```

In [39]:

```
# which company has maximum number of employess

df['Company Name'].value_counts()
```

Out[39]:

```
Tata Consultancy Services         271
Amazon                            184
Infosys                           169
Accenture                         150
Cognizant Technology Solutions    144
                                 ...
Talent Anywhere                     1
WisdmLabs                           1
Softdel                             1
Dentsu                              1
Nextgen Innovation Labs             1
Name: Company Name, Length: 11261, dtype: int64
```

Maximum employees work in TCS , Amazon , Infosys , Accenture and Cognizent.

In [40]:

```
# maximum employees works as which job title
df['Job Title'].value_counts()
```

Out[40]:

```
Software Development Engineer          2351
Android Developer                      2029
Software Development Engineer (SDE)    1614
Front End Developer                    1412
Test Engineer                          1314
                                       ... 
Java Andriod Developer                    1
Java Deceloper                            1
Java/J2EE Programmer                      1
Java SOA Developer                        1
Associate Web Developer                   1
Name: Job Title, Length: 1080, dtype: int64
```

Maximum employess work as SDE, Android devloper, Front End devloper and Test engineer.

In [41]:

```
df['Job Title'].unique()
```

Out[41]:

```
array(['Android Developer', 'Android Developer - Intern',
       'Android Developer - Contractor', ..., 'Web Developer Contracto
r',
       'Full Stack Web Developer', 'Associate Web Developer'],
      dtype=object)
```

In [42]:

```
# maximum employees works in which location
df['Location'].value_counts()
```

Out[42]:

```
Bangalore         8264
Hyderabad         4467
New Delhi         4176
Chennai           2458
Pune              2134
Mumbai             749
Kolkata            178
Madhya Pradesh     155
Kerala             108
Jaipur              81
Name: Location, dtype: int64
```

Maximum employees work Banglore, Hyderabad and New Delhi.

In [43]:

```
df['Job Roles'].value_counts()
```

Out[43]:

```
SDE           8183
Android       2945
Frontend      2163
Java          1858
Testing       1740
IOS           1631
Backend       1194
Web            999
Python         947
Database       865
Mobile         245
Name: Job Roles, dtype: int64
```

In [44]:

```
# distribution of employees on the basis of employment type

df['Employment Status'].value_counts()

```

Out[44]:

```
Full Time     20083
Intern         2106
Contractor      548
Trainee          33
Name: Employment Status, dtype: int64
```

In [45]:

```
df.isnull().sum()
```

Out[45]:

```
Rating               0
Company Name         0
Job Title            0
Salary               0
Salaries Reported    0
Location             0
Employment Status    0
Job Roles            0
dtype: int64
```

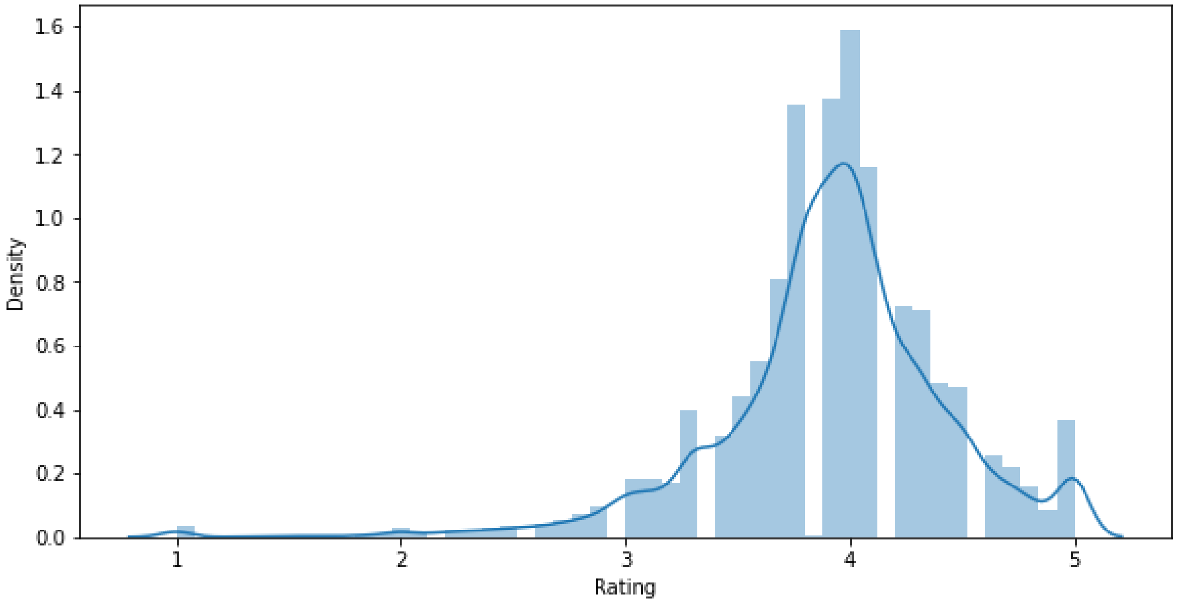
Visualizing insights of Features

In [46]:

```
# checking distribution of rating feature
plt.figure(figsize=(10,5))
sns.distplot(df['Rating'])
```

Out[46]:

```
<AxesSubplot:xlabel='Rating', ylabel='Density'>
```



In [47]:

```
df.columns
```

Out[47]:

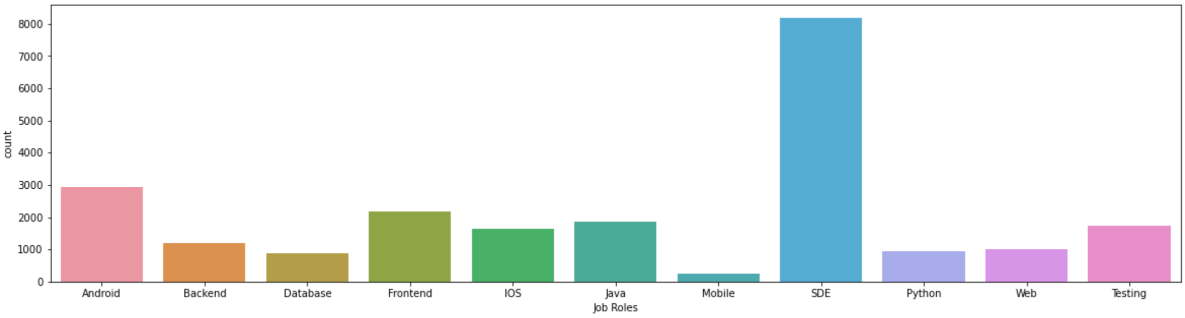
```
Index(['Rating', 'Company Name', 'Job Title', 'Salary', 'Salaries Repo
rted',
       'Location', 'Employment Status', 'Job Roles'],
      dtype='object')
```

In [48]:

```
plt.figure(figsize=(20,5))
sns.countplot(df['Job Roles'])

```

Out[48]:

```
<AxesSubplot:xlabel='Job Roles', ylabel='count'>
```

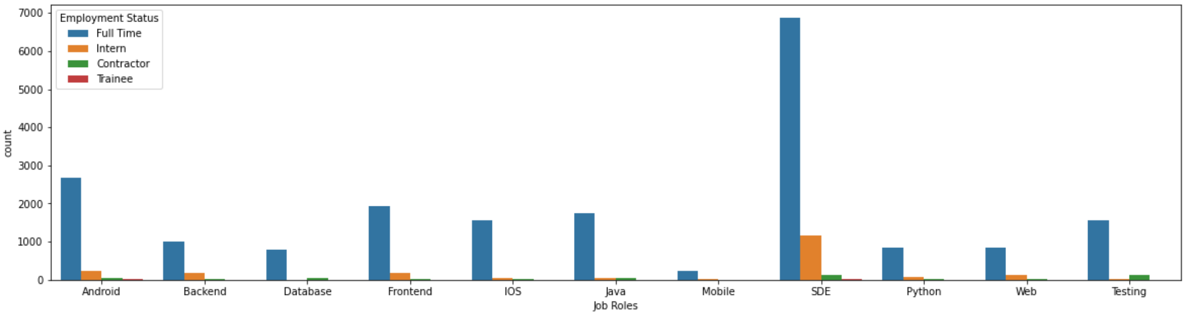


In [49]:

```
plt.figure(figsize=(20,5))
sns.countplot(df['Job Roles'],hue=df['Employment Status'])

```

Out[49]:

```
<AxesSubplot:xlabel='Job Roles', ylabel='count'>
```

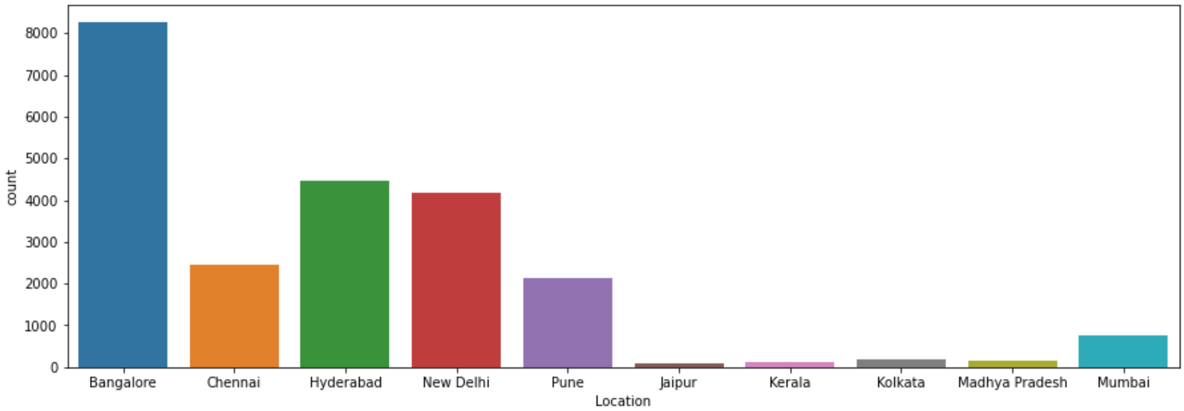


In [50]:

```
plt.figure(figsize=(15,5))
sns.countplot(df['Location'])

```

Out[50]:

```
<AxesSubplot:xlabel='Location', ylabel='count'>
```

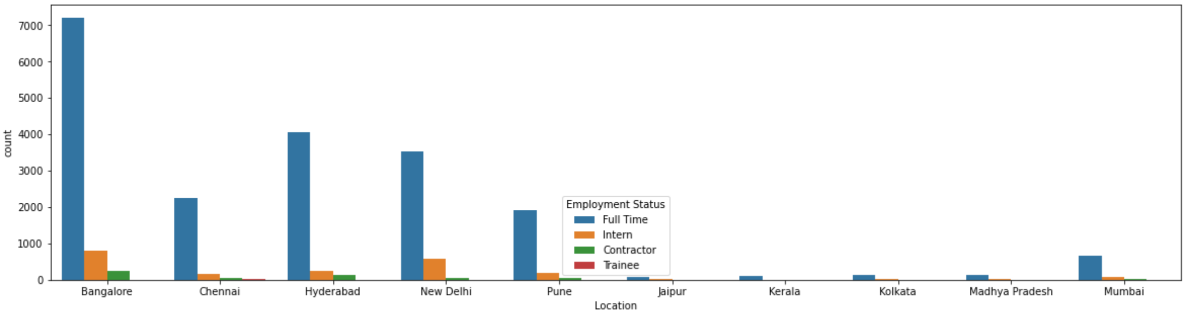


In [51]:

```
plt.figure(figsize=(20,5))
sns.countplot(df['Location'],hue=df['Employment Status'])

```

Out[51]:

```
<AxesSubplot:xlabel='Location', ylabel='count'>
```

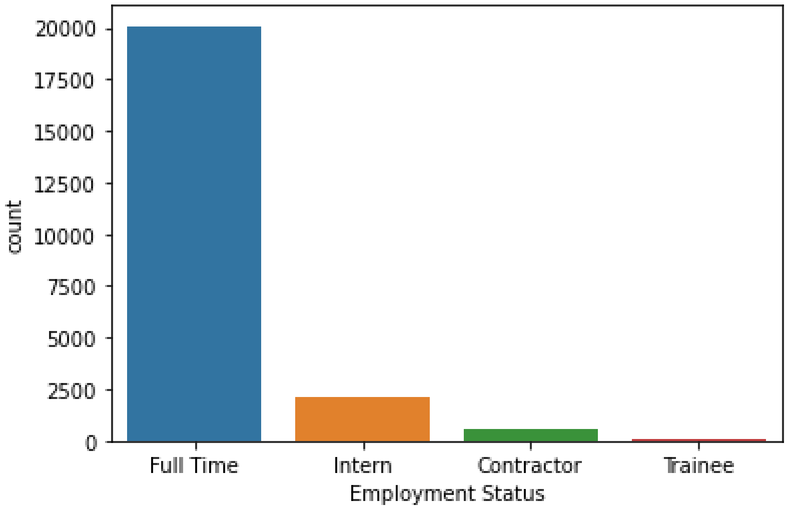


In [52]:

```
sns.countplot(df['Employment Status'])
```

Out[52]:

```
<AxesSubplot:xlabel='Employment Status', ylabel='count'>
```



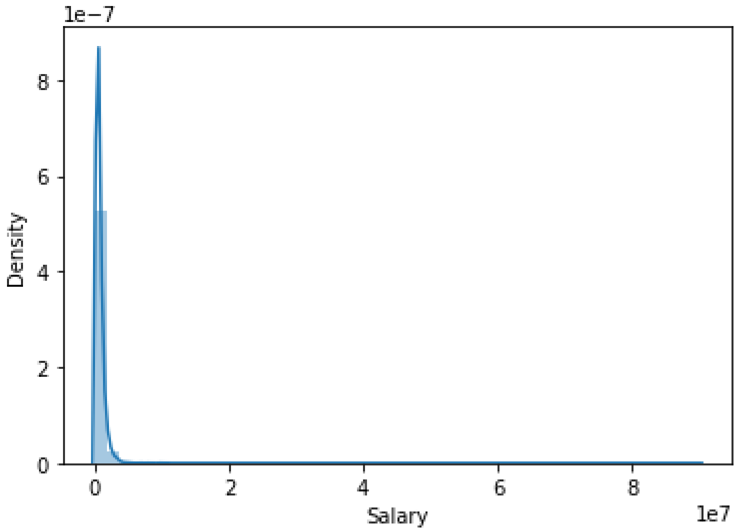
Peoples are more interested in getting Full time job as compared to intern , contrater and trainee.

In [53]:

```
sns.distplot(df['Salary'])
```

Out[53]:

```
<AxesSubplot:xlabel='Salary', ylabel='Density'>
```



In [54]:

```
df['Salaries Reported'].value_counts()
```

Out[54]:

```
1      18206
2       2401
3        789
4        382
5        228
       ...  
39         1
162        1
361        1
135        1
100        1
Name: Salaries Reported, Length: 82, dtype: int64
```

In [55]:

```
# Top 20 companies with 5.0 ratings.

df[['Company Name','Rating']].sort_values('Rating',ascending=False).head(20)
```

Out[55]:

	Company Name	Rating
20326	Nagalakshmi Solutions	5.0
4938	MyDBOPS	5.0
22486	Felicity Software Solutions	5.0
7807	Samsact	5.0
22483	Tihalt Technologies	5.0
3148	Loco	5.0
17351	Parth Universal	5.0
3144	Botrecruits Software	5.0
3142	PC Financial	5.0
22479	Hawx Media	5.0
15374	Random Math	5.0
8930	Ascendz HR Solutions	5.0
22478	Webtrackers4u	5.0
3139	Business Toys	5.0
3138	COVIAM	5.0
8519	Loyakk	5.0
3134	Winkl	5.0
21938	Technobuk	5.0
3124	AppRaam Labs	5.0
575	StraightDrive Softlab	5.0

In [56]:

```
highest_salary_job = df.nlargest(5,['Salary'])
highest_salary_job
```

Out[56]:

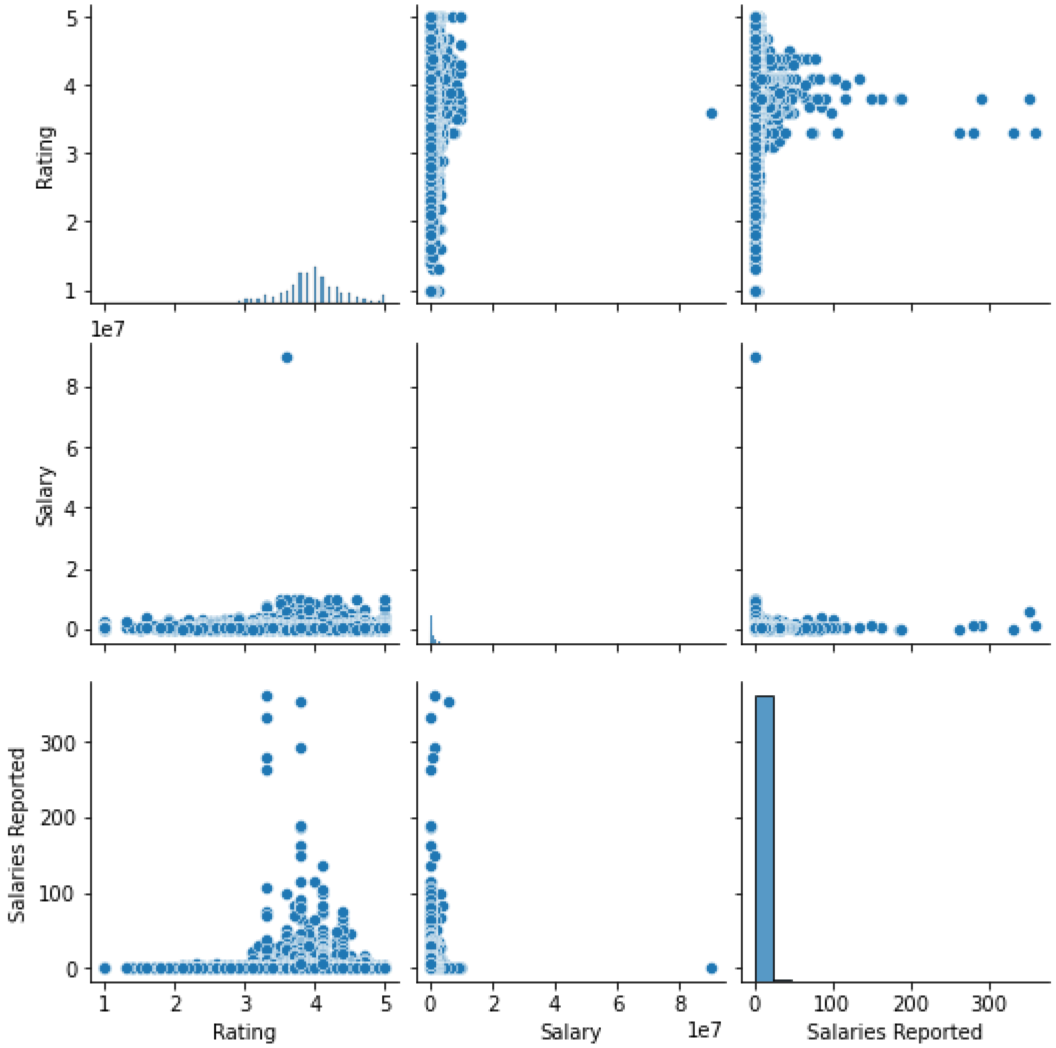
	Rating	Company Name	Job Title	Salary	Salaries Reported	Location	Employment Status	Job Roles
18635	3.6	Thapar University	Software Development Engineer (SDE)	90000000	1	New Delhi	Full Time	SDE
4471	3.8	Concentrix	Oracle Database Administrator	10000000	1	Bangalore	Full Time	Database
7121	3.5	Koru UX Design	Senior Front End Developer	10000000	1	Pune	Full Time	Frontend
9260	3.6	OASYS Cybernetics	Senior Java Developer	10000000	1	Chennai	Full Time	Java
5819	3.7	Nityo Infotech	Lead UI Designer, Magento Front-end Developer	9900000	1	Bangalore	Full Time	Frontend

In [57]:

```
sns.pairplot(df)
```

Out[57]:

```
<seaborn.axisgrid.PairGrid at 0x7fa4d8b280a0>
```

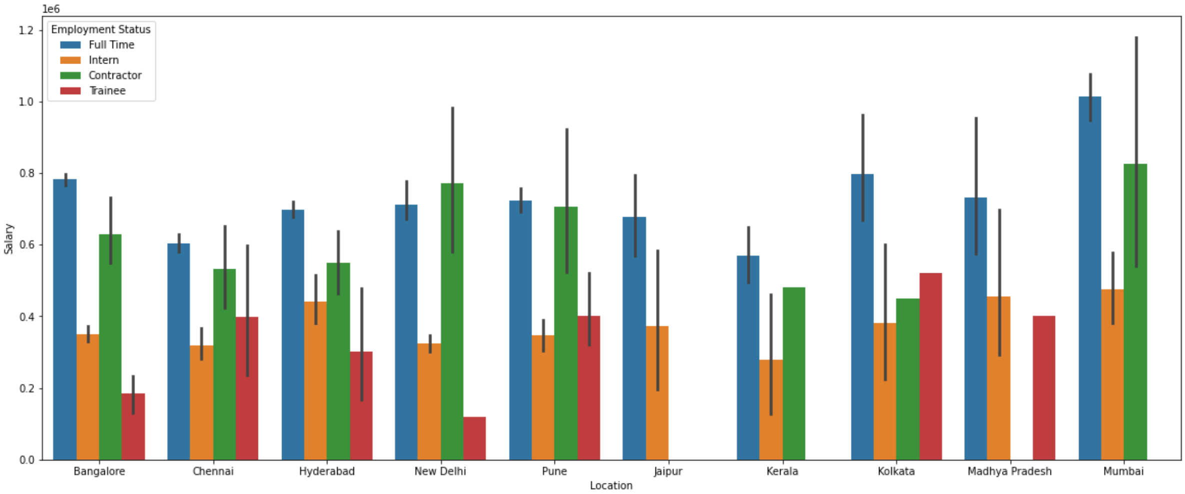


In [58]:

```
plt.figure(figsize=(20,8))
sns.barplot(data=df,x='Location',y='Salary',hue=df['Employment Status'])
```

Out[58]:

```
<AxesSubplot:xlabel='Location', ylabel='Salary'>
```

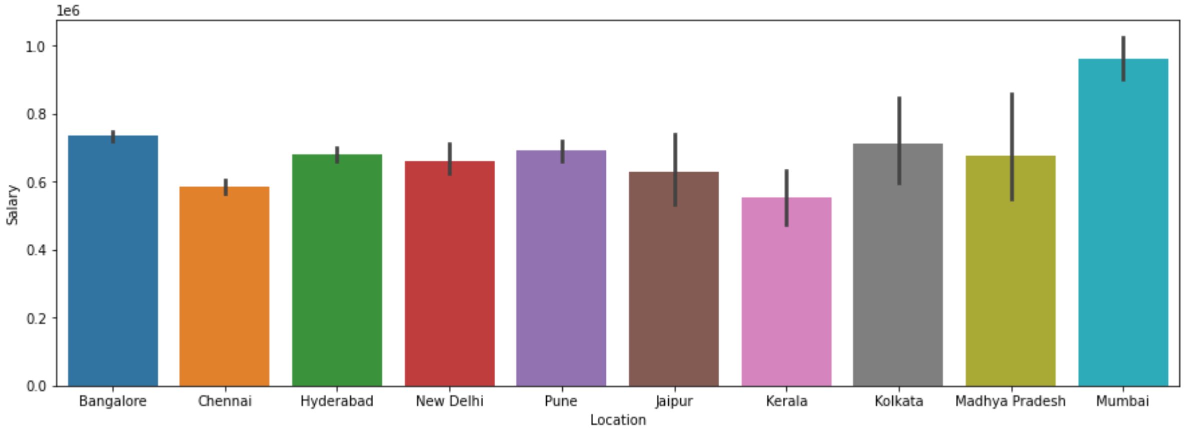


In [59]:

```
plt.figure(figsize=(15,5))
sns.barplot(data=df,x='Location',y='Salary')
```

Out[59]:

```
<AxesSubplot:xlabel='Location', ylabel='Salary'>
```



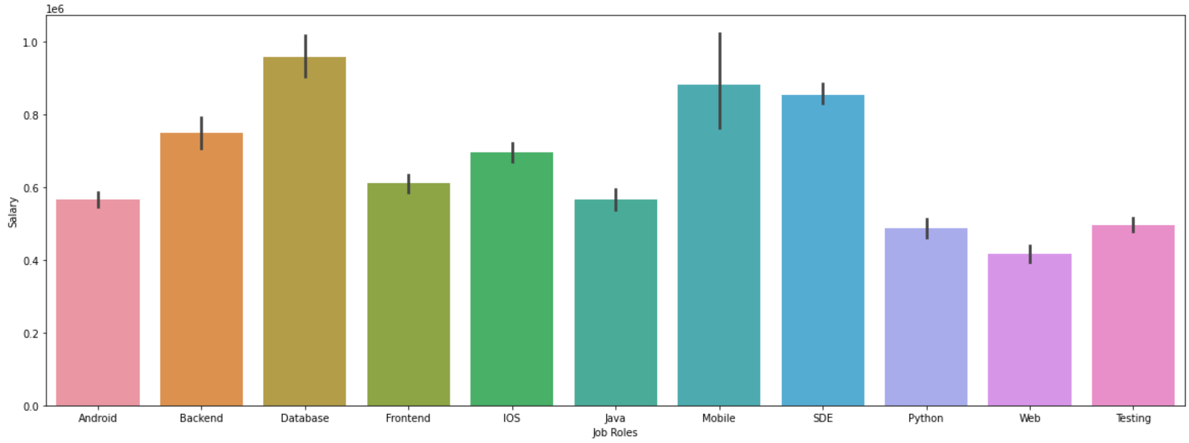
We can clearly see that people working in mumbai get more salary rather than other location.

In [60]:

```
plt.figure(figsize=(20,7))
sns.barplot(data=df,x='Job Roles',y='Salary')
```

Out[60]:

```
<AxesSubplot:xlabel='Job Roles', ylabel='Salary'>
```

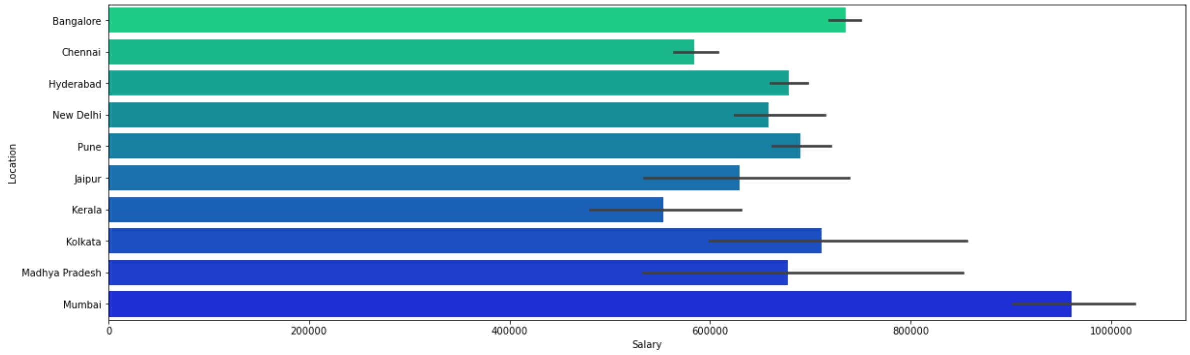


Instead maximum people work as SDE but they can't pay much by companies. We can see Database job role payed more rather than other job roles.

Location wise salary

In [61]:

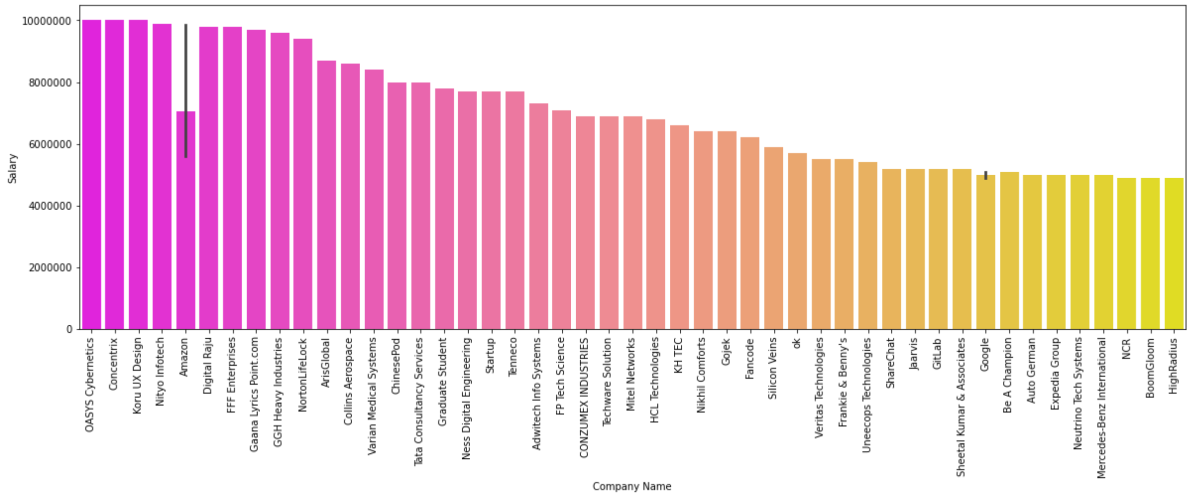
```
plt.figure(figsize = (20, 6))
plt.ticklabel_format(style = 'plain')
sns.barplot(x = df["Salary"], y = df["Location"], palette = "winter_r");
```



COMPANY SALARY

In [62]:

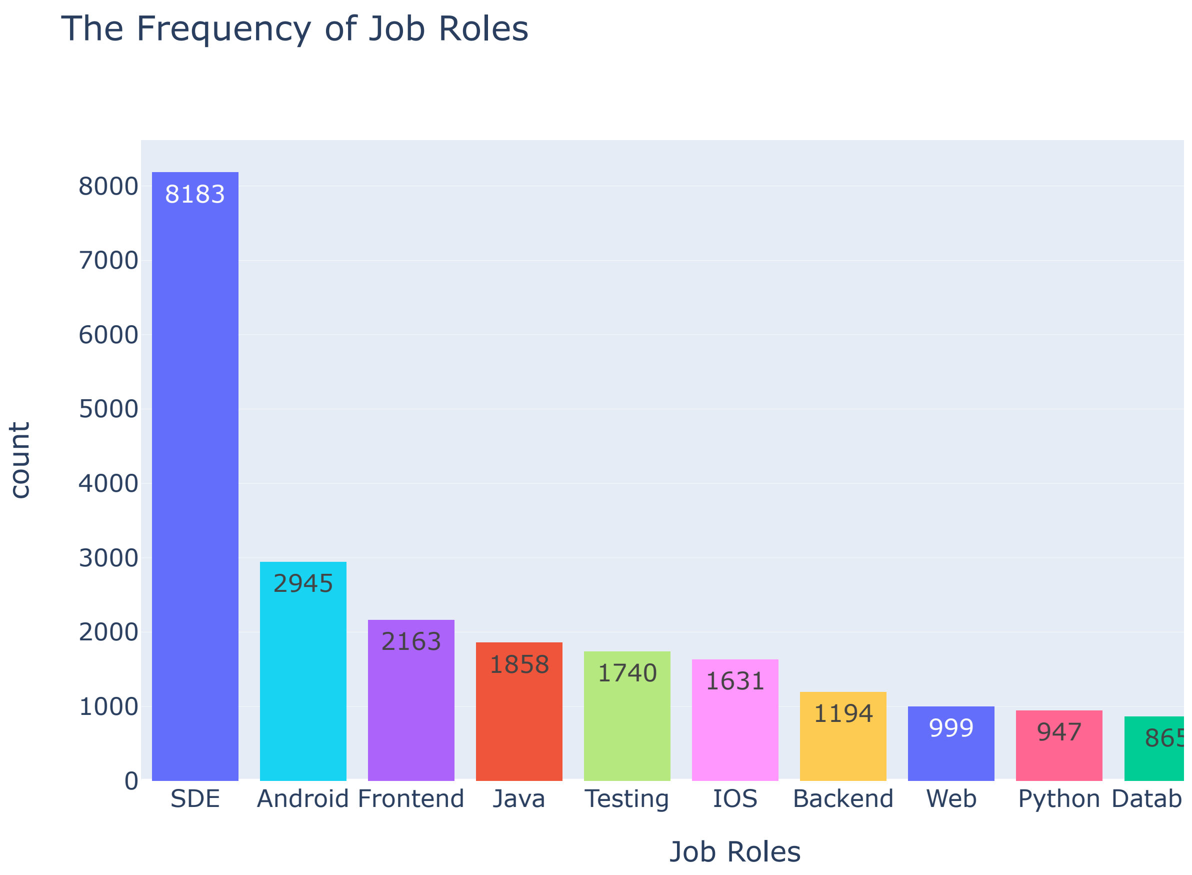
```
plt.figure(figsize = (20, 6))
plt.xticks(rotation = 90)
plt.ticklabel_format(style = 'plain')
df.sort_values("Salary", axis = 0, ascending = False, inplace = True)
sns.barplot(x = df["Company Name"][1:51],
    y = df["Salary"][1:51],
    palette = "spring");
```



This Graph shows the frequency of the Job Roles.

In [64]:

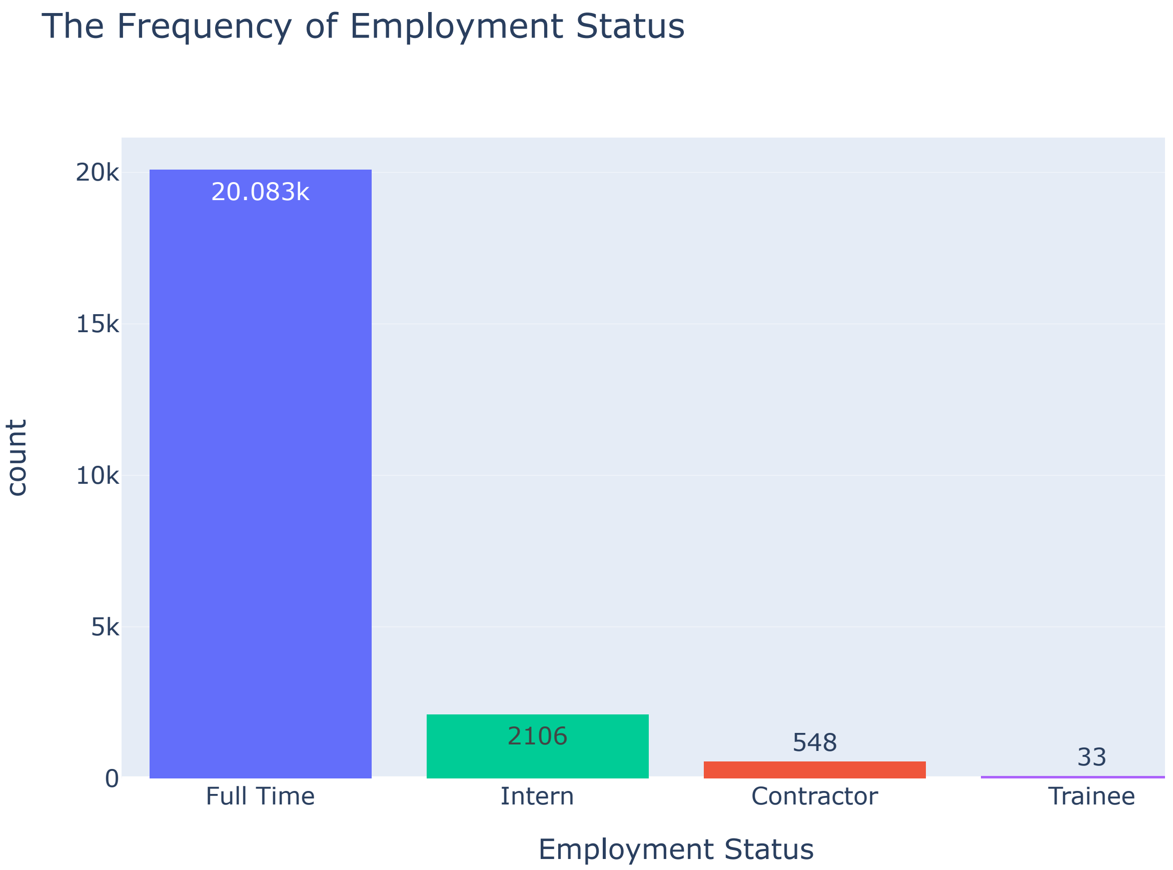
```
# Graph(1)
df1 = px.histogram(df, x='Job Roles', color='Job Roles', title="The Frequency of
df1.update_layout(xaxis={'categoryorder' : 'total descending'})
df1.show()
```



This Graph shows the frequency of the Employment Status.

In [65]:

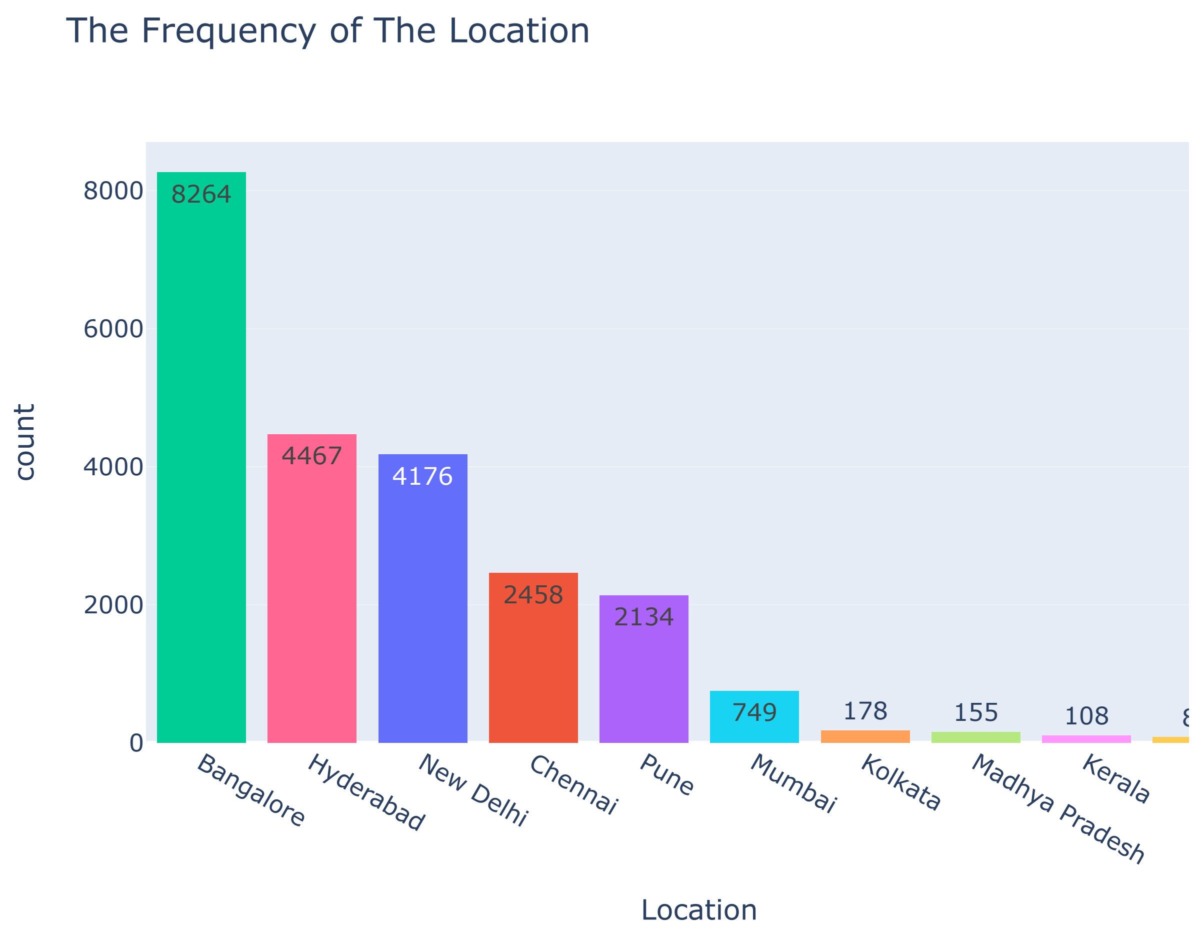
```
# Graph(2)
df2 = px.histogram(df, x='Employment Status', color='Employment Status', title='
df2.update_layout(xaxis={'categoryorder' : 'total descending'})
df2.show()
```



This Graph shows the frequency of the Location.

In [66]:

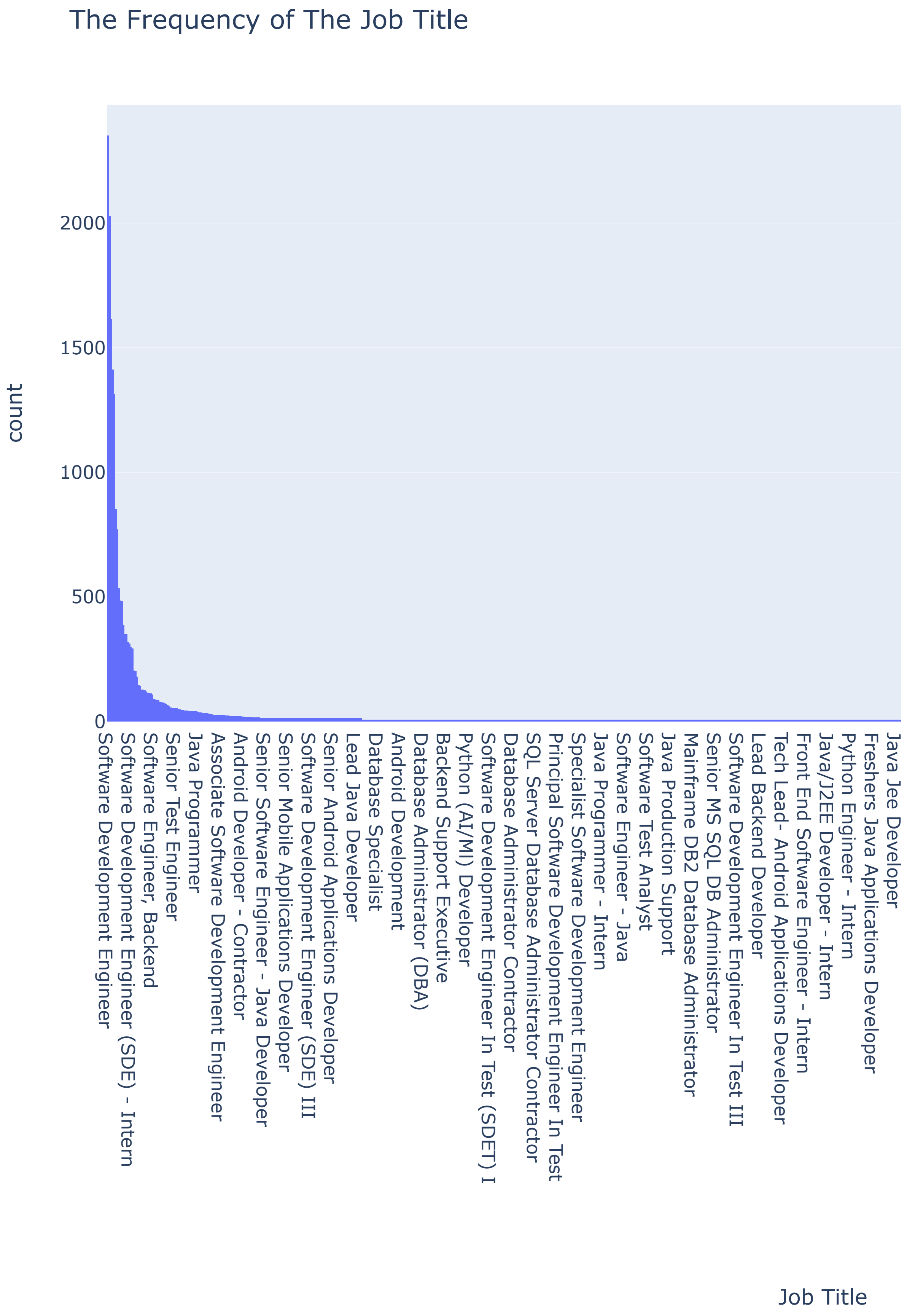
```
# Graph(3)
df3 = px.histogram(df, x='Location', color='Location', title="The Frequency of T
df3.update_layout(xaxis={'categoryorder' : 'total descending'})
df3.show()

```



This Graph shows the frequency of the job title.

In [68]:

```
# Graph(4)
df4 = px.histogram(df, x='Job Title', title="The Frequency of The Job Title", wi
df4.update_layout(xaxis={'categoryorder' : 'total descending'})
df4.show()
```

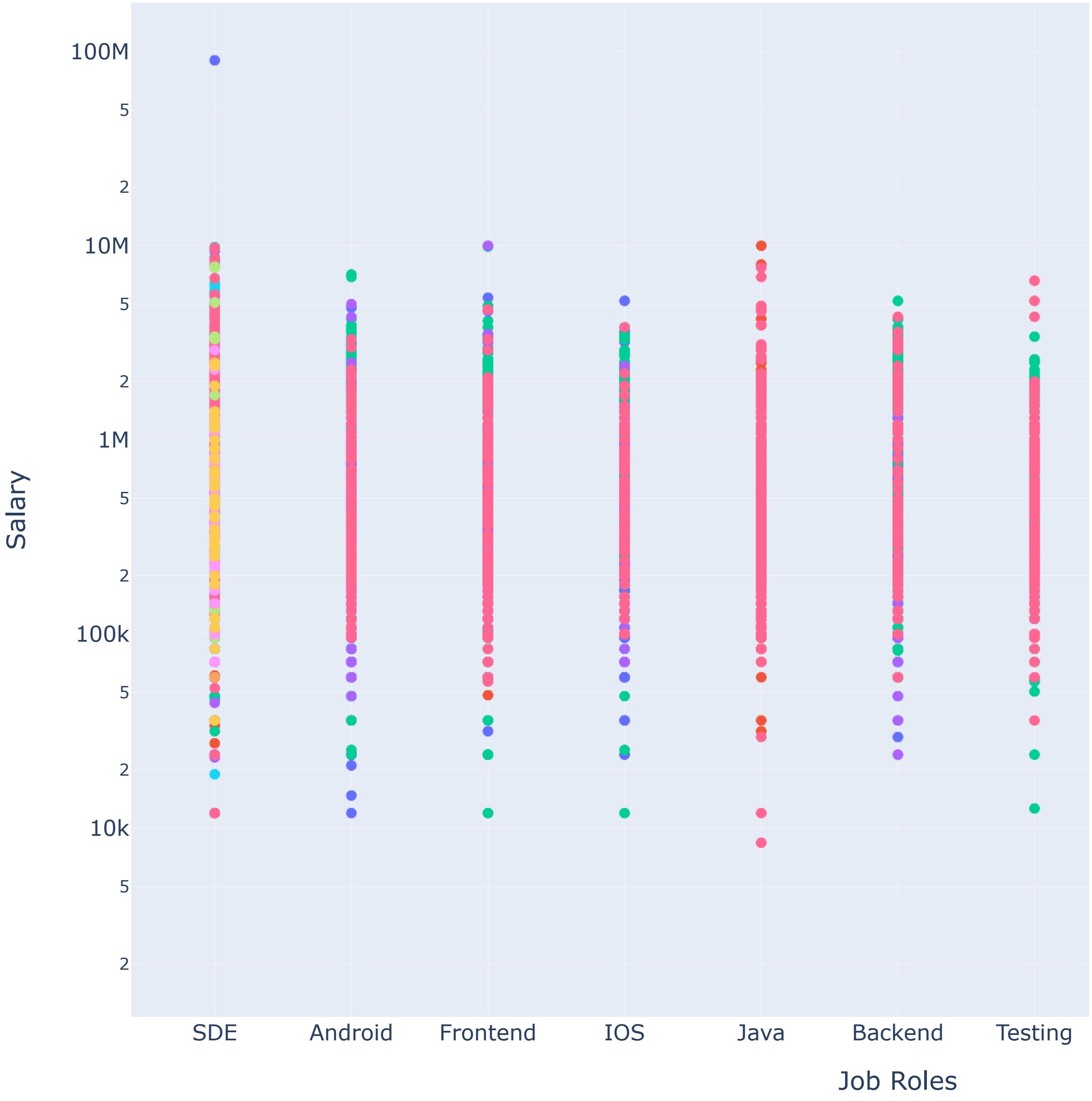


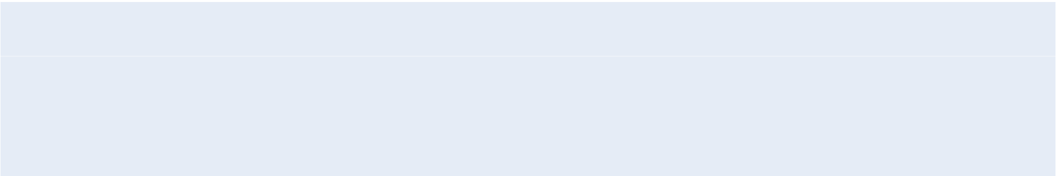
This Graph indicates the salaries for every 'job role' and the distribution by location:

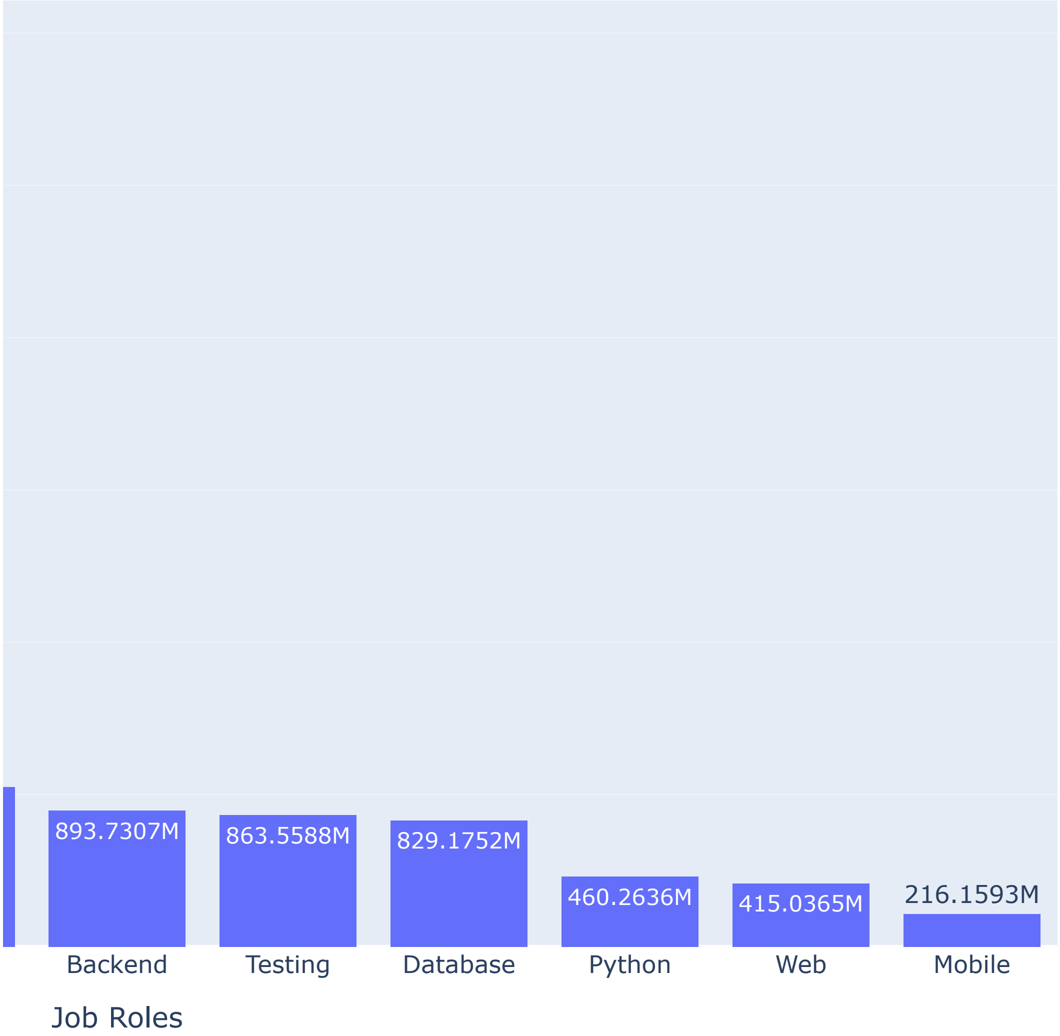
We can easily control the graph to check the data properly. We can notice the sum of salaries in job role 'SDE' (Software Development Engineer) is the highest one. We can notice there are different salaries according to location

In [70]:

```
_frame=df, x='Job Roles', y='Salary', color='Location', width=1100, height=700, log_y
s={'categoryorder' : 'total descending'})


ata_frame=df, x='Job Roles', y='Salary', width=1100, height=700, text_auto=True)
xis={'categoryorder' : 'total descending'})
```



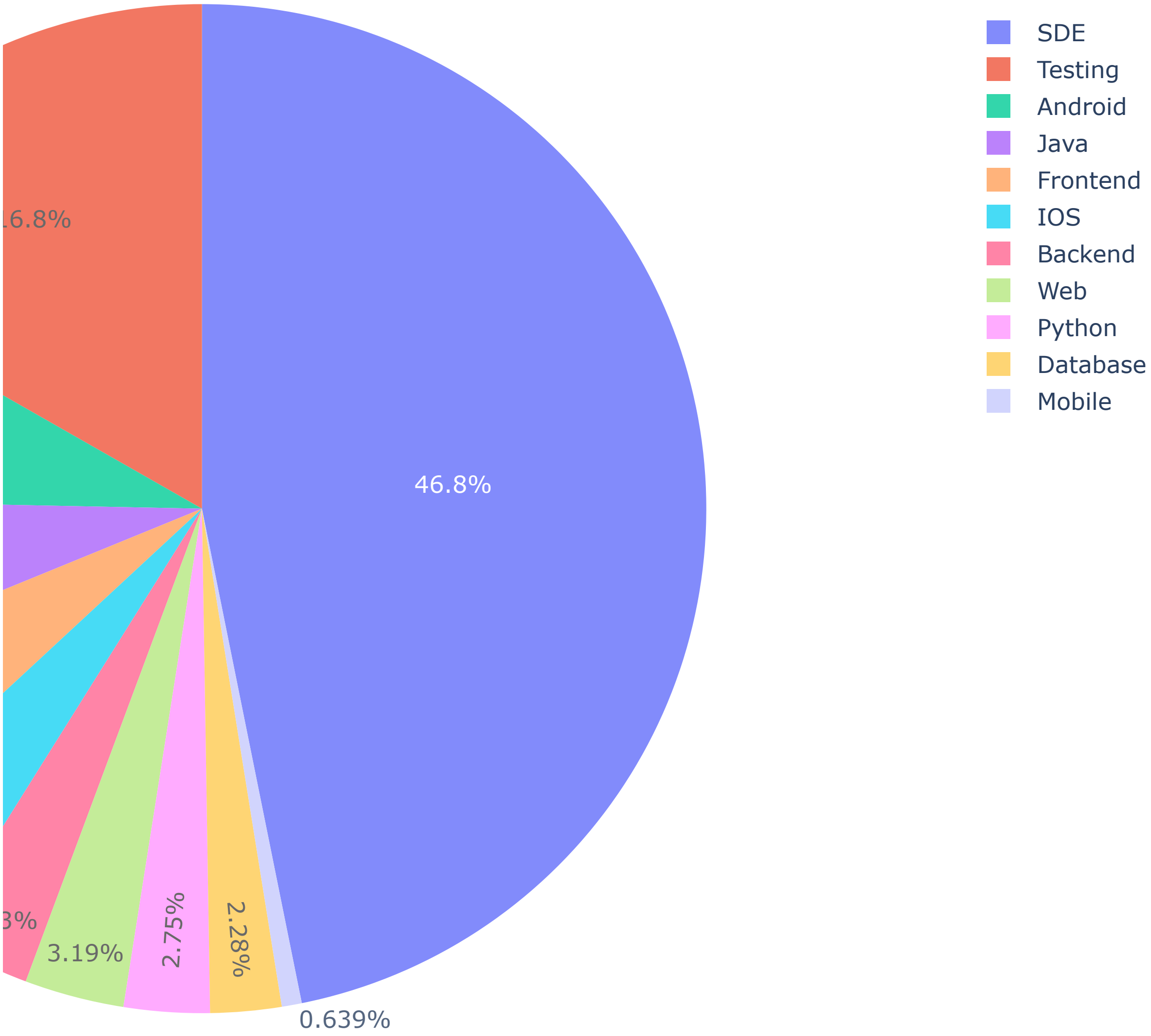




This graph shows the percentage of Job roles and how many employees reported their salary.

In [72]:

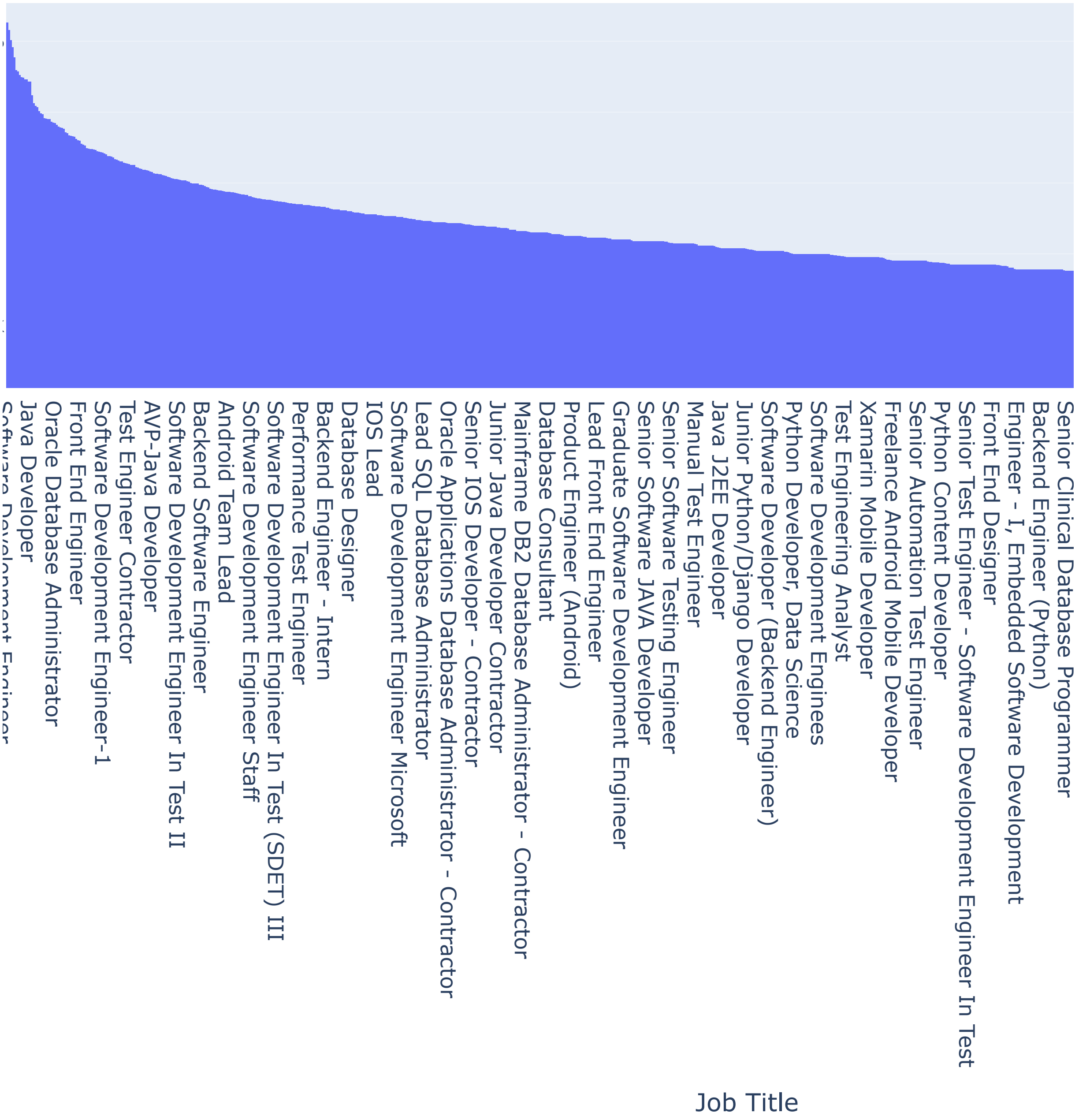
```
oles',values='Salaries Reported', width=1000, height=700, opacity=0.8, title="Role")
```



This Graph indicates the sum of salaries for every 'Job Title'.

In [74]:

```
# Graph(7)
df7 = px.histogram(data_frame=df, x='Job Title', y='Salary', width=1000, height=
df7.update_layout(xaxis={'categoryorder' : 'total descending'})
df7.show()

```



In [76]:

```
df['Salary'].describe()
```

Out[76]:

```
count    2.277000e+04
mean     6.953872e+05
std      8.843990e+05
min      2.112000e+03
25%      3.000000e+05
50%      5.000000e+05
75%      9.000000e+05
max      9.000000e+07
Name: Salary, dtype: float64
```

In [77]:

```python
# The top ten high salaries
df.sort_values(by='Salary', ascending=False).head(10)
```

Out[77]:

	Rating	Company Name	Job Title	Salary	Salaries Reported	Location	Employment Status	Job Roles
18635	3.6	Thapar University	Software Development Engineer (SDE)	90000000	1	New Delhi	Full Time	SDE
7121	3.5	Koru UX Design	Senior Front End Developer	10000000	1	Pune	Full Time	Frontenc
9260	3.6	OASYS Cybernetics	Senior Java Developer	10000000	1	Chennai	Full Time	Java
4471	3.8	Concentrix	Oracle Database Administrator	10000000	1	Bangalore	Full Time	Database
5819	3.7	Nityo Infotech	Lead UI Designer, Magento Front-end Developer	9900000	1	Bangalore	Full Time	Frontenc
16062	3.8	Amazon	Software Development Engineer (SDE)	9850000	1	Kolkata	Full Time	SDE
18654	4.3	Digital Raju	Software Development Engineer (SDE)	9800000	1	New Delhi	Full Time	SDE
16619	4.2	FFF Enterprises	Non Software Development Engineer	9800000	1	Mumbai	Full Time	SDE
15568	5.0	Gaana Lyrics Point.com	Software Development Engineer (SDE) II	9700000	1	Hyderabad	Full Time	SDE
10832	4.6	GGH Heavy Industries	Best Buy Mobile Sales Associate	9600000	1	Bangalore	Full Time	Mobile

In [79]:

```
# The lowest ten salaries
df.sort_values(by='Salary').head(10)
```

Out[79]:

	Rating	Company Name	Job Title	Salary	Salaries Reported	Location	Employment Status	Job Roles
22563	2.6	Keeves Technologies	Web Developer - Intern	2112	1	Bangalore	Intern	Web
9937	3.7	Virtusa	Junior Java Developer	8448	5	Hyderabad	Full Time	Java
10316	3.9	Awign Enterprises	Senior Java Developer Contractor	12000	1	Hyderabad	Contractor	Java
21912	4.2	JAVA	Web Developer	12000	1	Bangalore	Full Time	Web
18422	3.4	BharatPe	Software Development Engineer (SDE)	12000	1	New Delhi	Full Time	SDE
15076	3.8	XYZ	Software Development Engineer (SDE) - Intern	12000	1	Hyderabad	Intern	SDE
16838	4.4	Excel Engineering Services	Senior Software Development Engineer	12000	1	Mumbai	Full Time	SDE
18429	4.0	....	Software Development Engineer (SDE)	12000	1	New Delhi	Full Time	SDE
2472	3.5	Acmatics Technologies	Android Developer	12000	1	New Delhi	Full Time	Android
22028	3.9	Yes Bank	Web Developer	12000	1	Bangalore	Full Time	Web

we are going to split out the data frame according to the job role column, and do some graphs for separated data.

In [82]:

```
# splitting the data into a small dataframe
sde = (df.loc[df['Job Roles'] == 'SDE'])
android = (df.loc[df['Job Roles'] == 'Android'])
frontend = (df.loc[df['Job Roles'] == 'Frontend'])
ios = (df.loc[df['Job Roles'] == 'IOS'])
java = (df.loc[df['Job Roles'] == 'Java'])
backend = (df.loc[df['Job Roles'] == 'Backend'])
testing = (df.loc[df['Job Roles'] == 'Testing'])
database = (df.loc[df['Job Roles'] == 'Database'])
py = (df.loc[df['Job Roles'] == 'Python'])
web = (df.loc[df['Job Roles'] == 'Web'])
mobile = (df.loc[df['Job Roles'] == 'Mobile'])
```

Graph(8) has (11) graphs inside it to indicate:

First of all The sum of salaries for each job role.

In addition, it indicates which cities have that 'job role' indeed, As we can notice some cities don't have some 'job roles'.

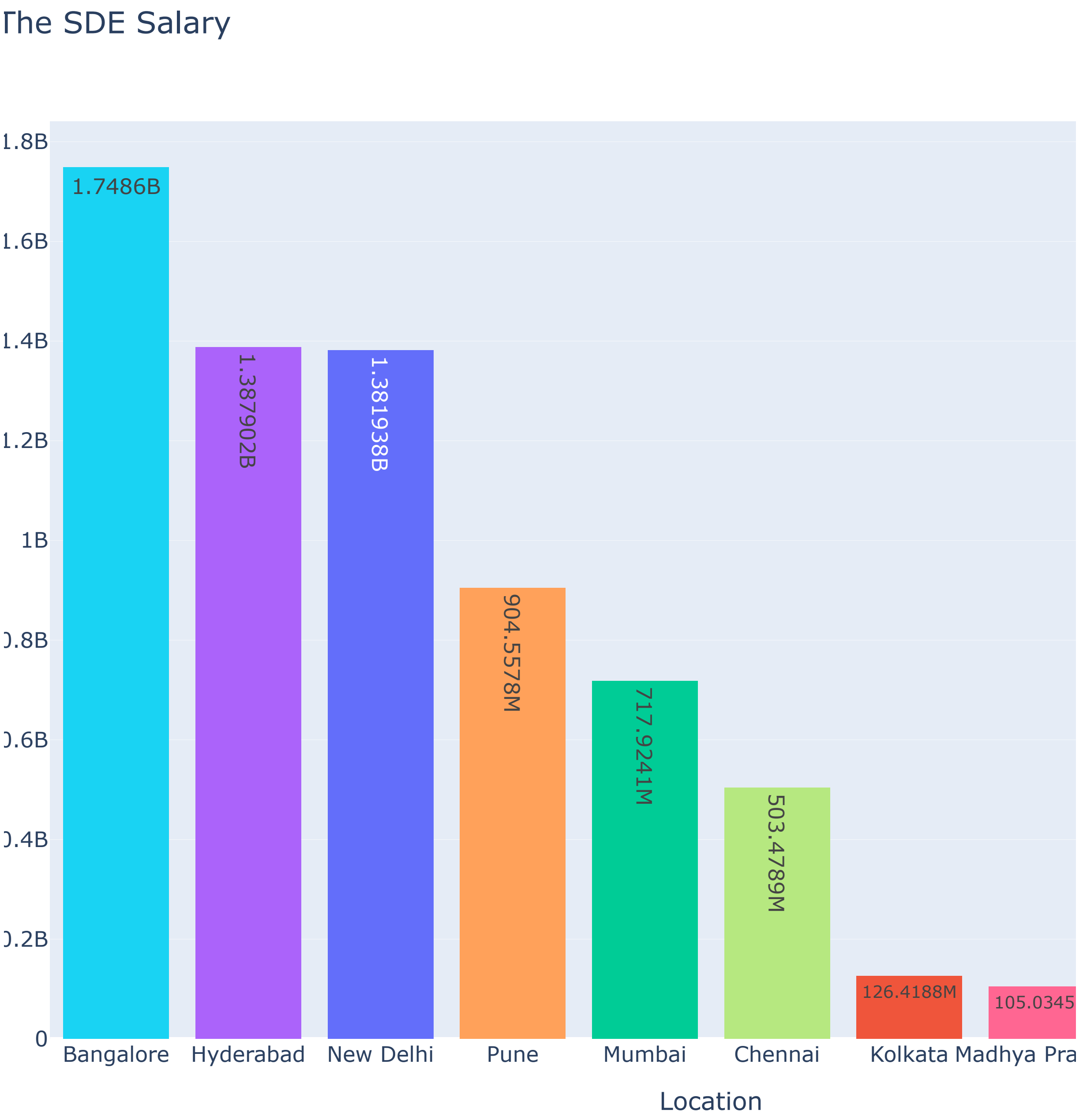
Moreover, we can notice that 'Bangalore' is the city that has jobs for every job role except 'Java'

According to This graph, you can check what is your job role and select which city has the highest salary and if it's available or unavailable.

Finally, feel free to leave a comment and tell me what can you extract from this graph!?!

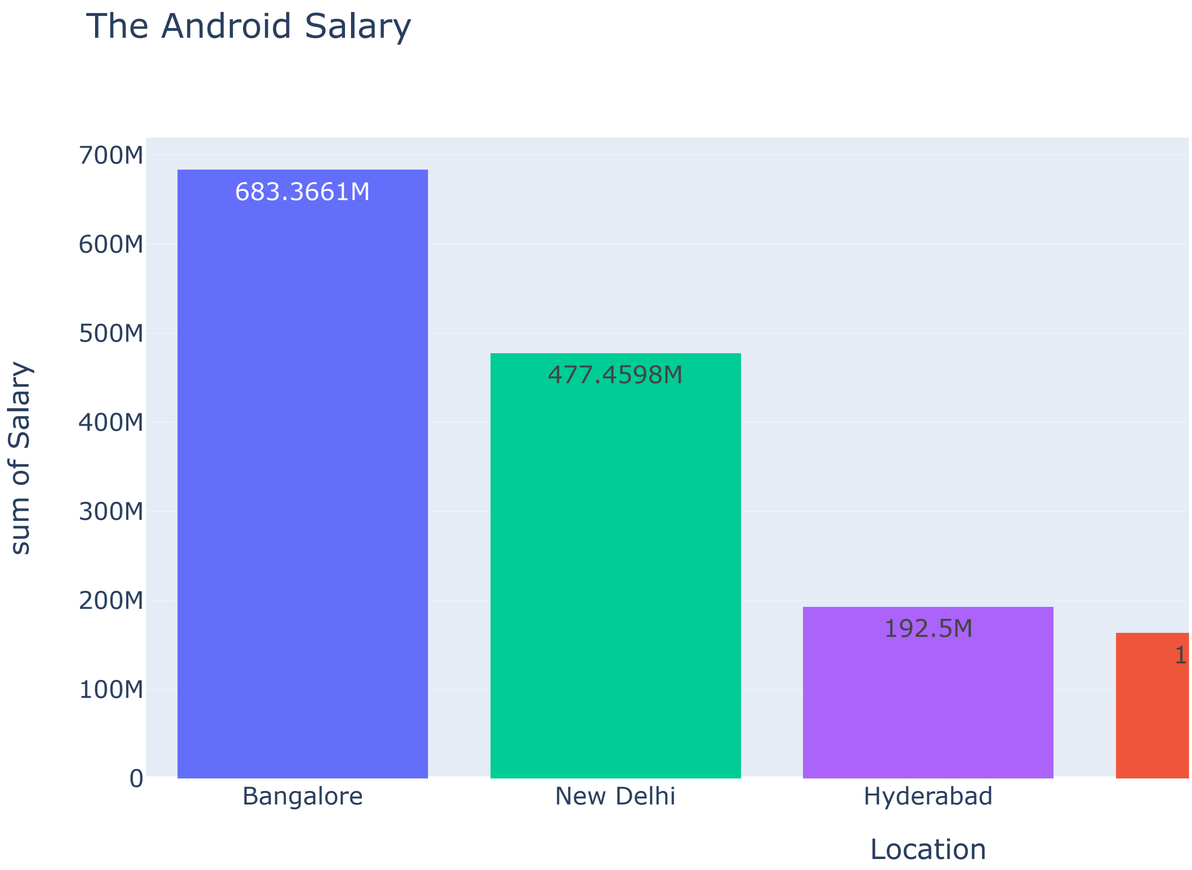
In [83]:

```
Graph(8, 1)
de_df = px.histogram(data_frame=sde, x='Location', y='Salary', width=1000, height=70
de_df.update_layout(xaxis={'categoryorder' : 'total descending'})
de_df.show()
```



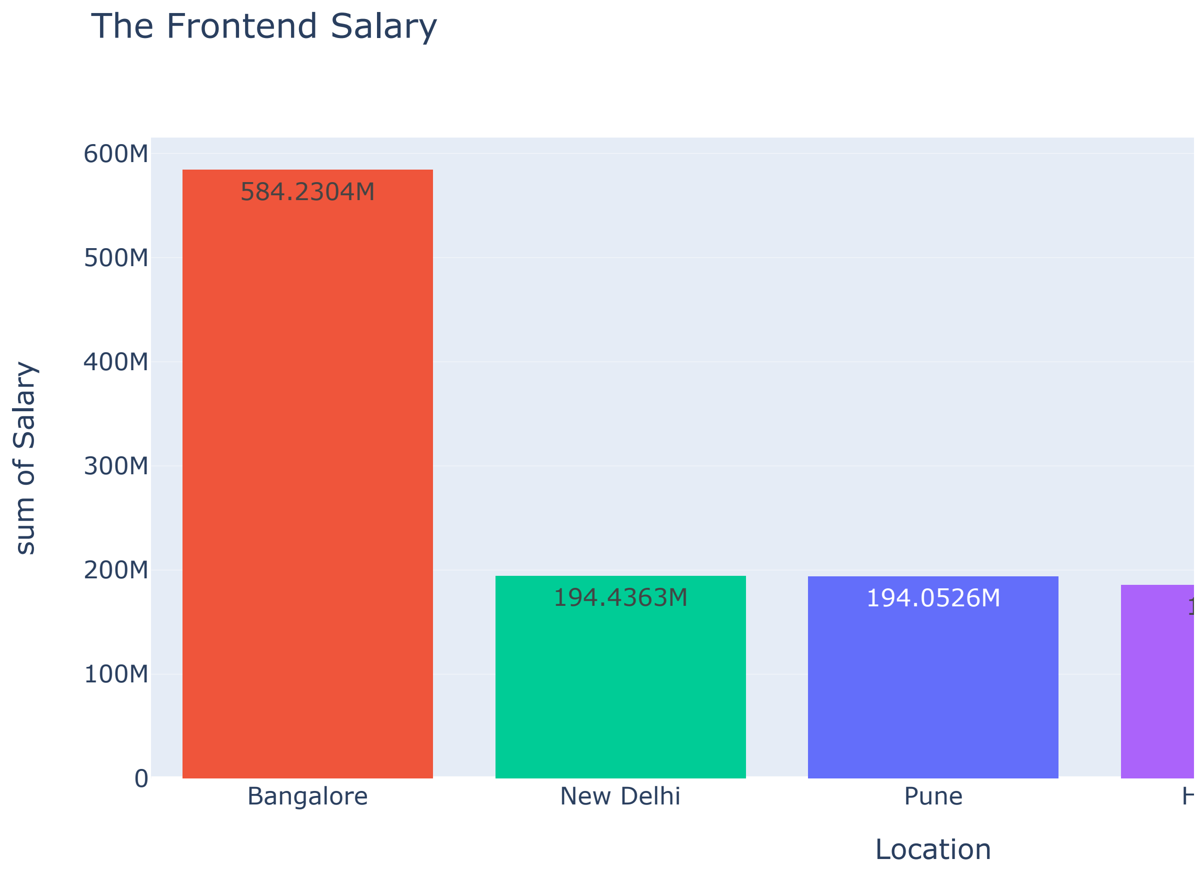
In [84]:

```
x='Location', y='Salary', width=1000, height=500, title="The Android Salary", text_a
' : 'total descending'})
```



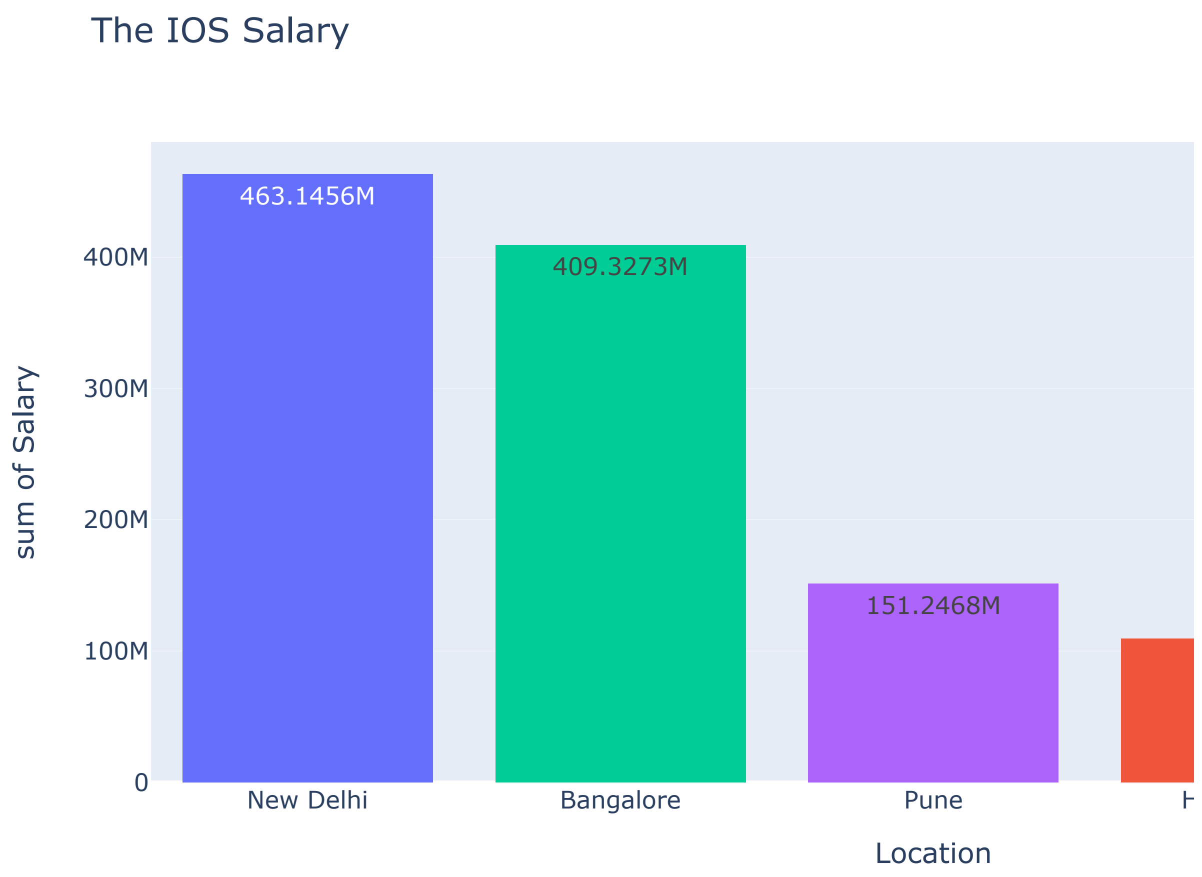
In [85]:

```
ontend, x='Location', y='Salary', width=1000, height=500, title="The Frontend Salary"
ryorder' : 'total descending'})
```



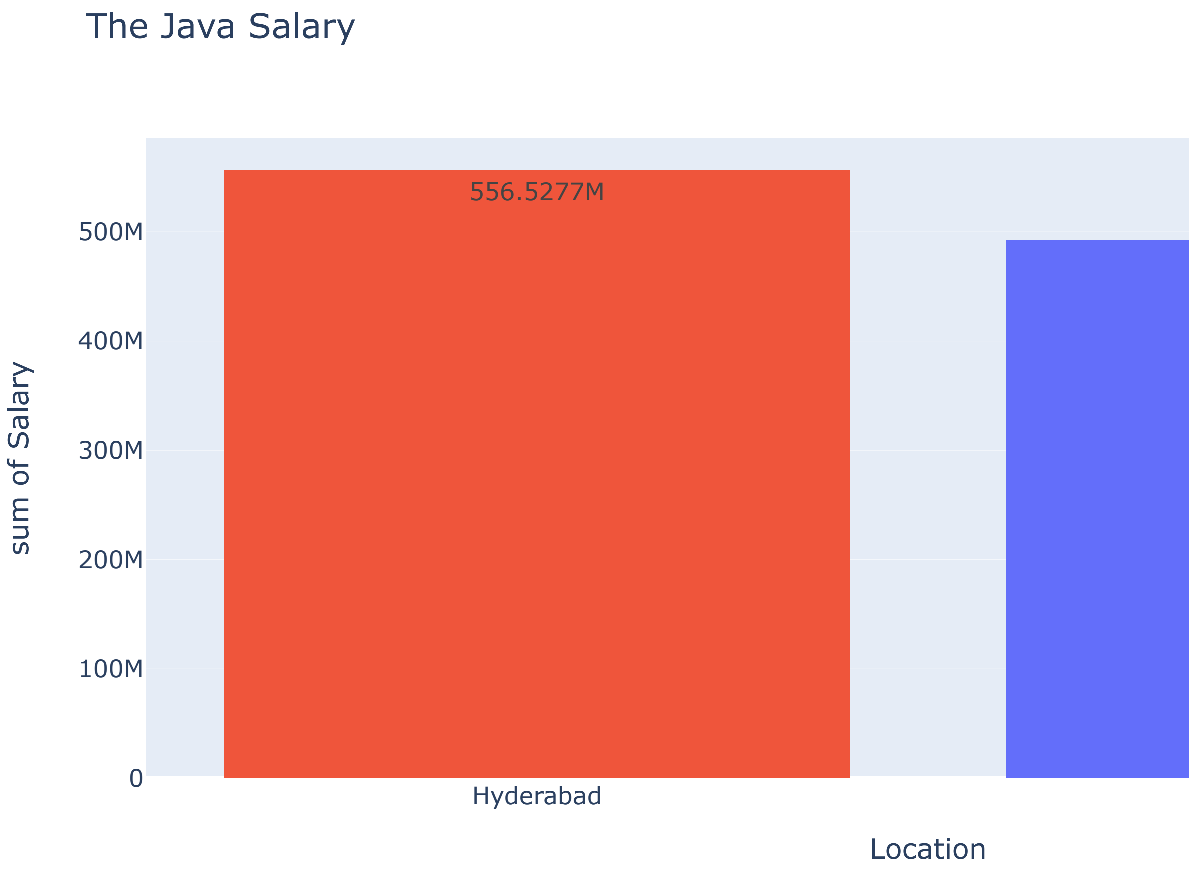
In [86]:

```
tion', y='Salary', width=1000, height=500, title="The IOS Salary", text_auto=True, co
 'total descending'})
```



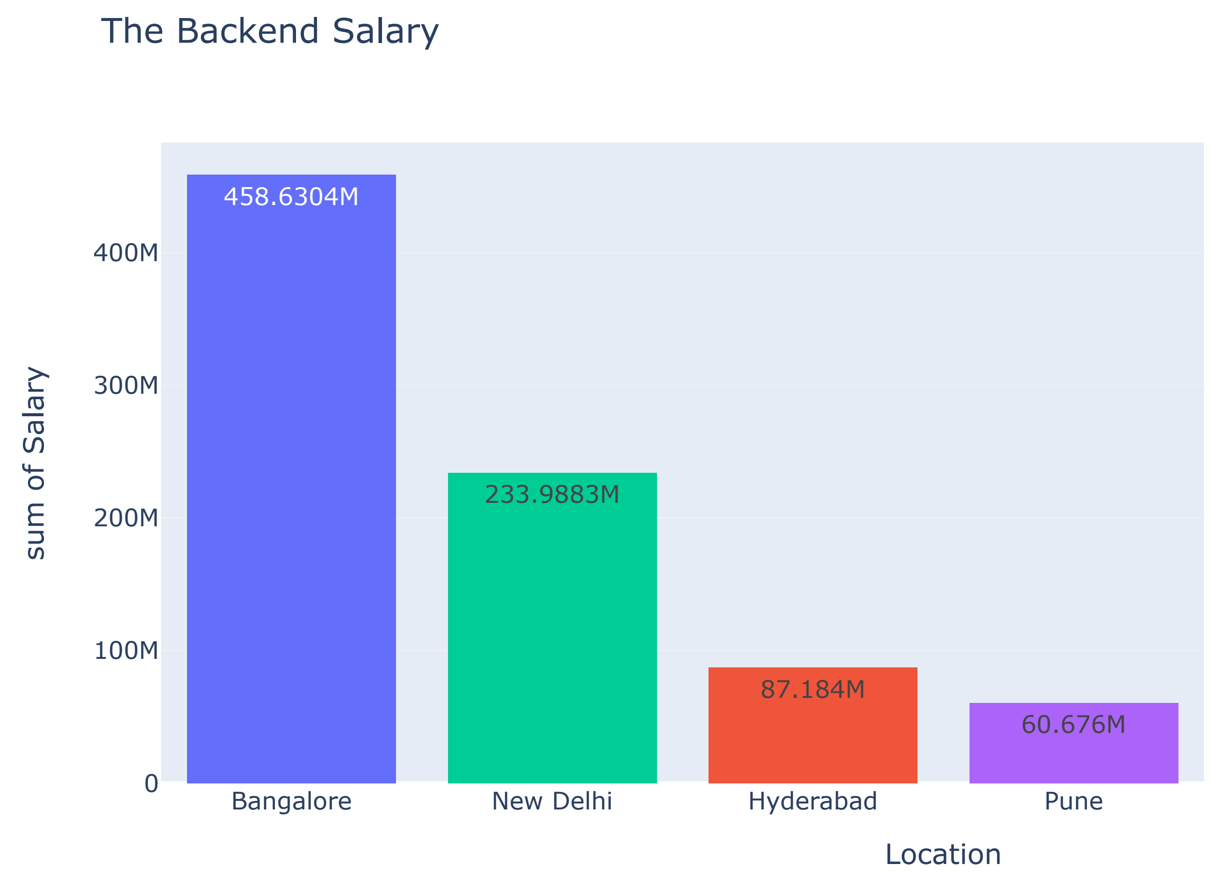
In [87]:

```
8, 5)
= px.histogram(data_frame=java, x='Location', y='Salary', width=1000, height=500, ti
update_layout(xaxis={'categoryorder' : 'total descending'})
show()
```



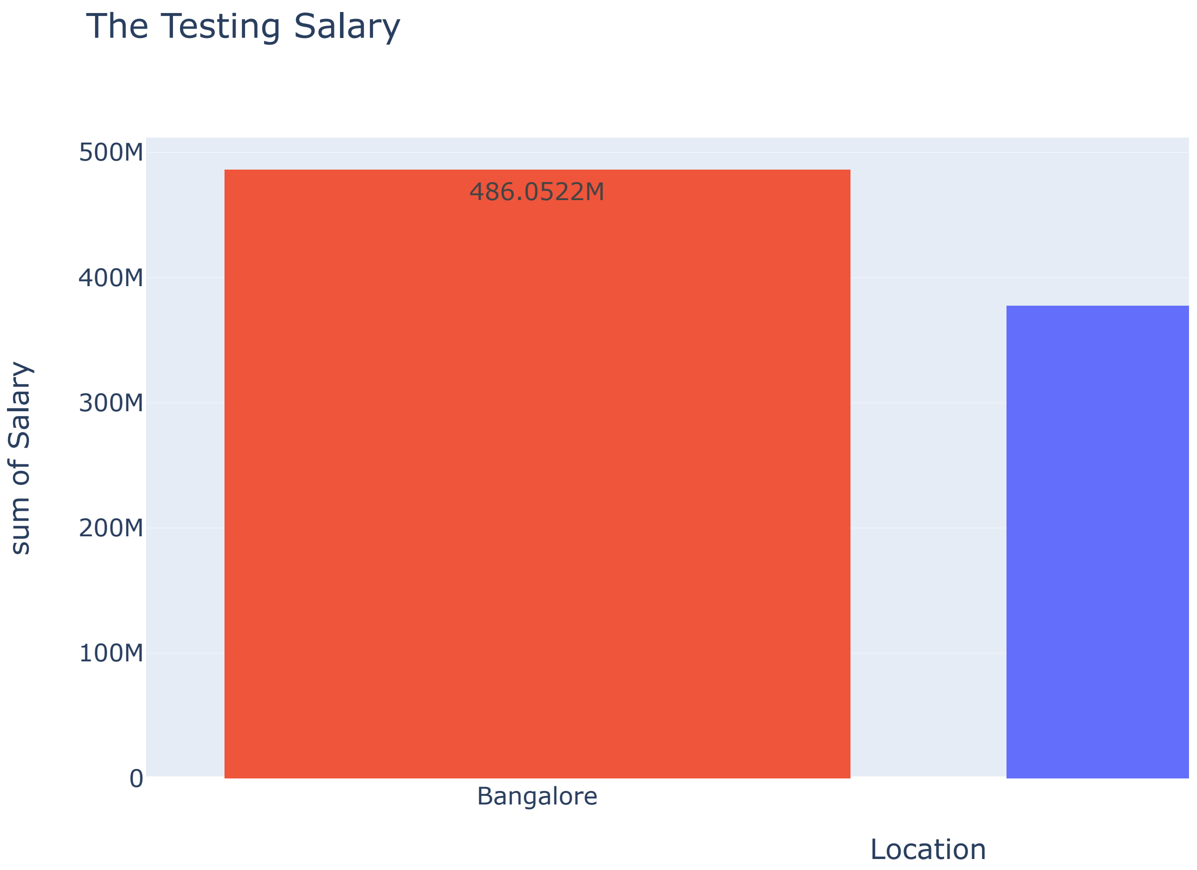
In [88]:

```
# Graph(8, 6)
backend_df = px.histogram(data_frame=backend, x='Location', y='Salary', width=1000,
backend_df.update_layout(xaxis={'categoryorder' : 'total descending'})
backend_df.show()
```



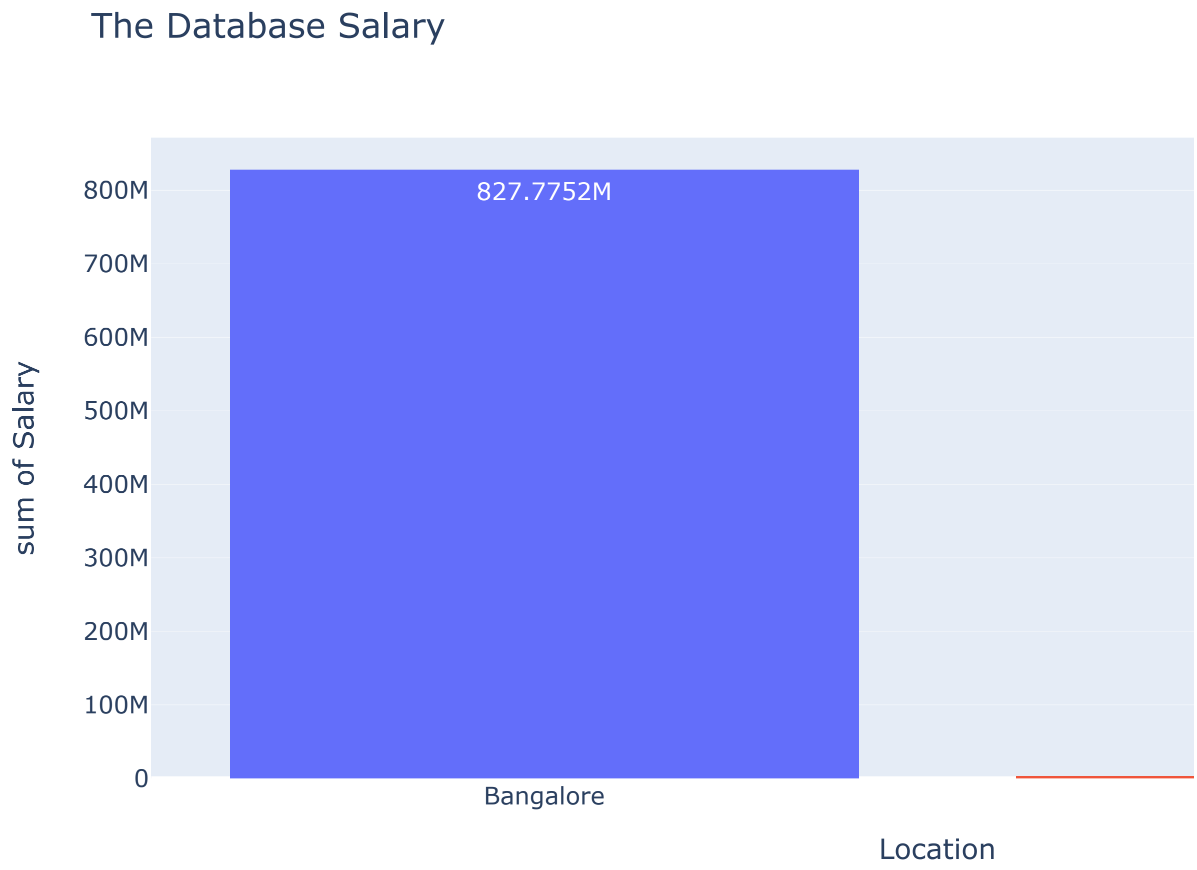
In [89]:

```
7)
= px.histogram(data_frame=testing, x='Location', y='Salary', width=1000, height=500,
update_layout(xaxis={'categoryorder' : 'total descending'})
show()
```



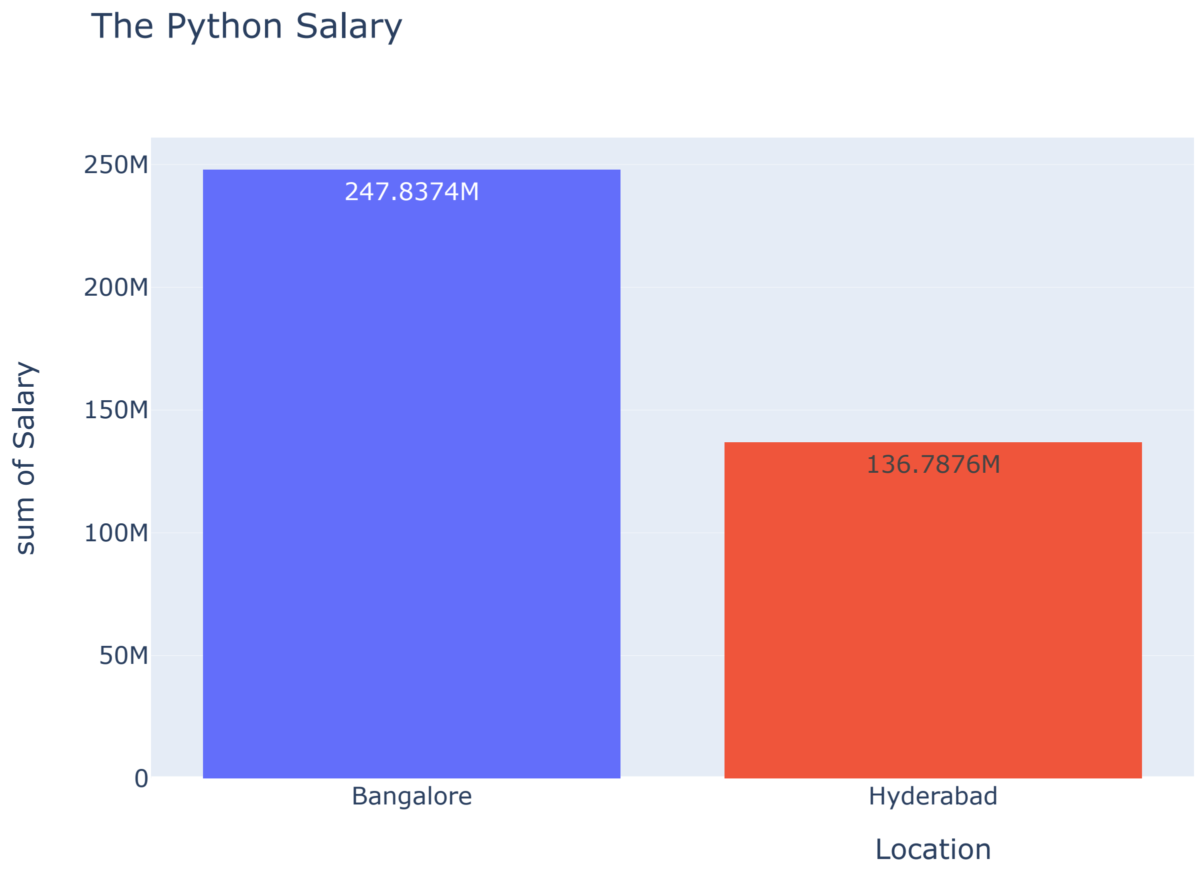
In [90]:

```
.histogram(data_frame=database, x='Location', y='Salary', width=1000, height=500, ti
te_layout(xaxis={'categoryorder' : 'total descending'})
()
```



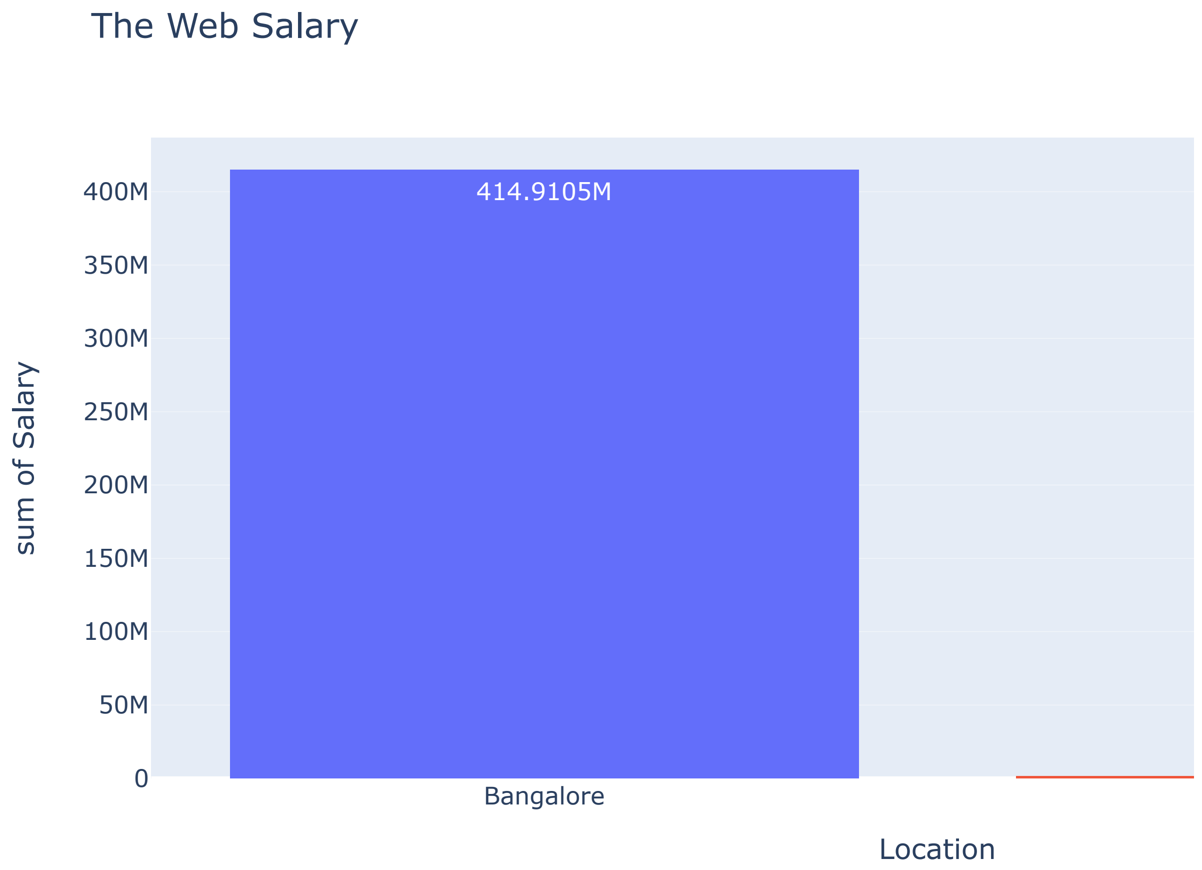
In [91]:

```
9)
histogram(data_frame=py, x='Location', y='Salary', width=1000, height=500, title="The
e_layout(xaxis={'categoryorder' : 'total descending'})
)
```



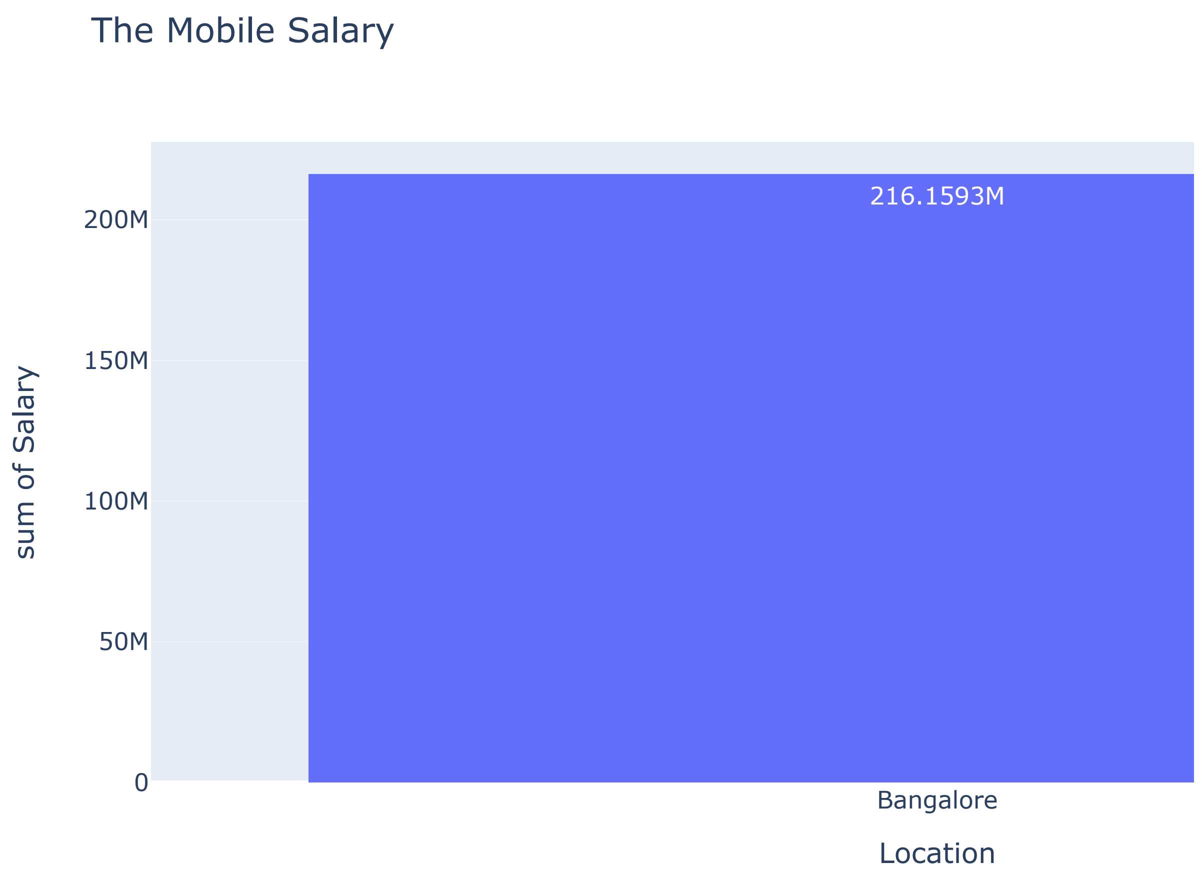
In [92]:

```
# Graph(8, 10)
web_df = px.histogram(data_frame=web, x='Location', y='Salary', width=1000, heig
web_df.update_layout(xaxis={'categoryorder' : 'total descending'})
web_df.show()
```



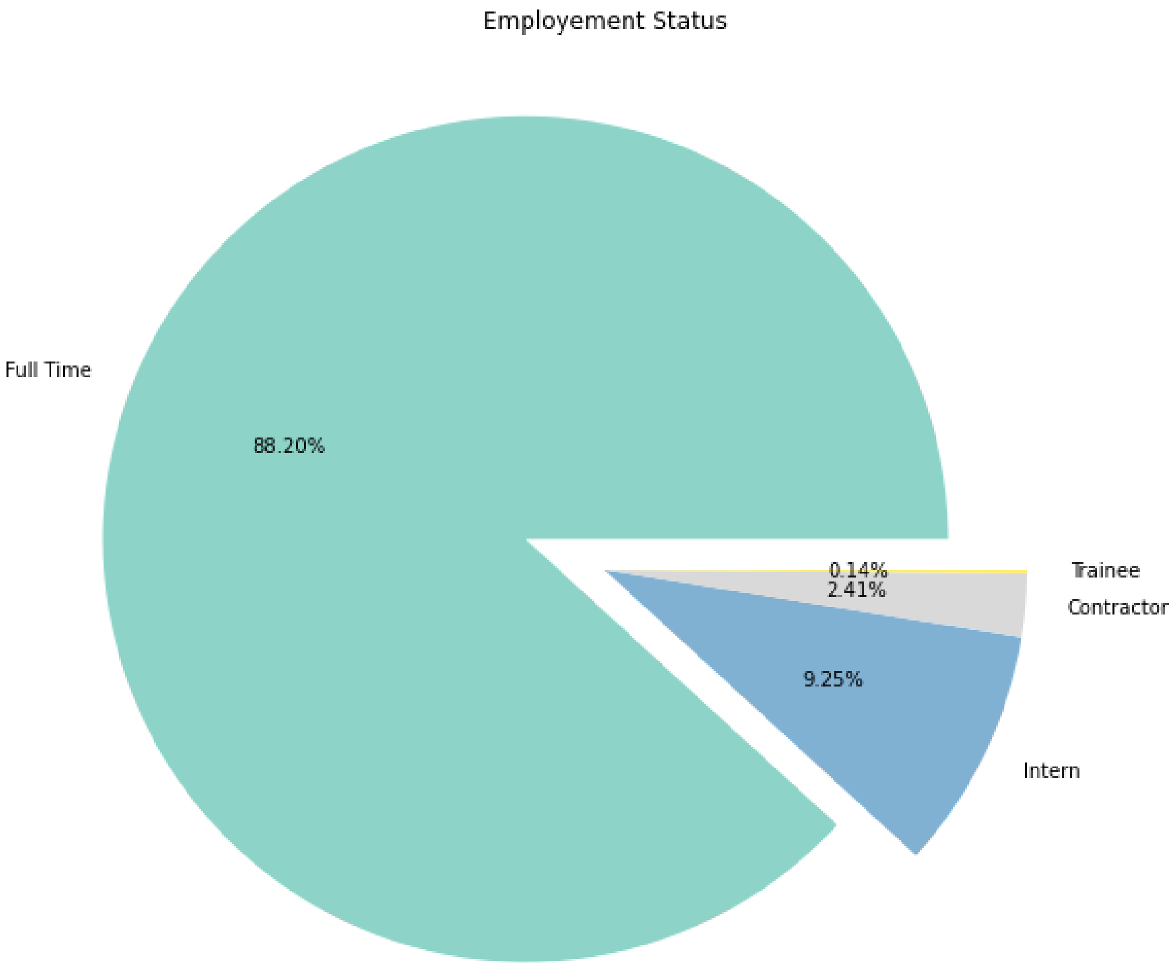
In [93]:

```
Graph(8, 11)
obile_df = px.histogram(data_frame=mobile, x='Location', y='Salary', width=1000, hei
obile_df.update_layout(xaxis={'categoryorder' : 'total descending'})
obile_df.show()
```



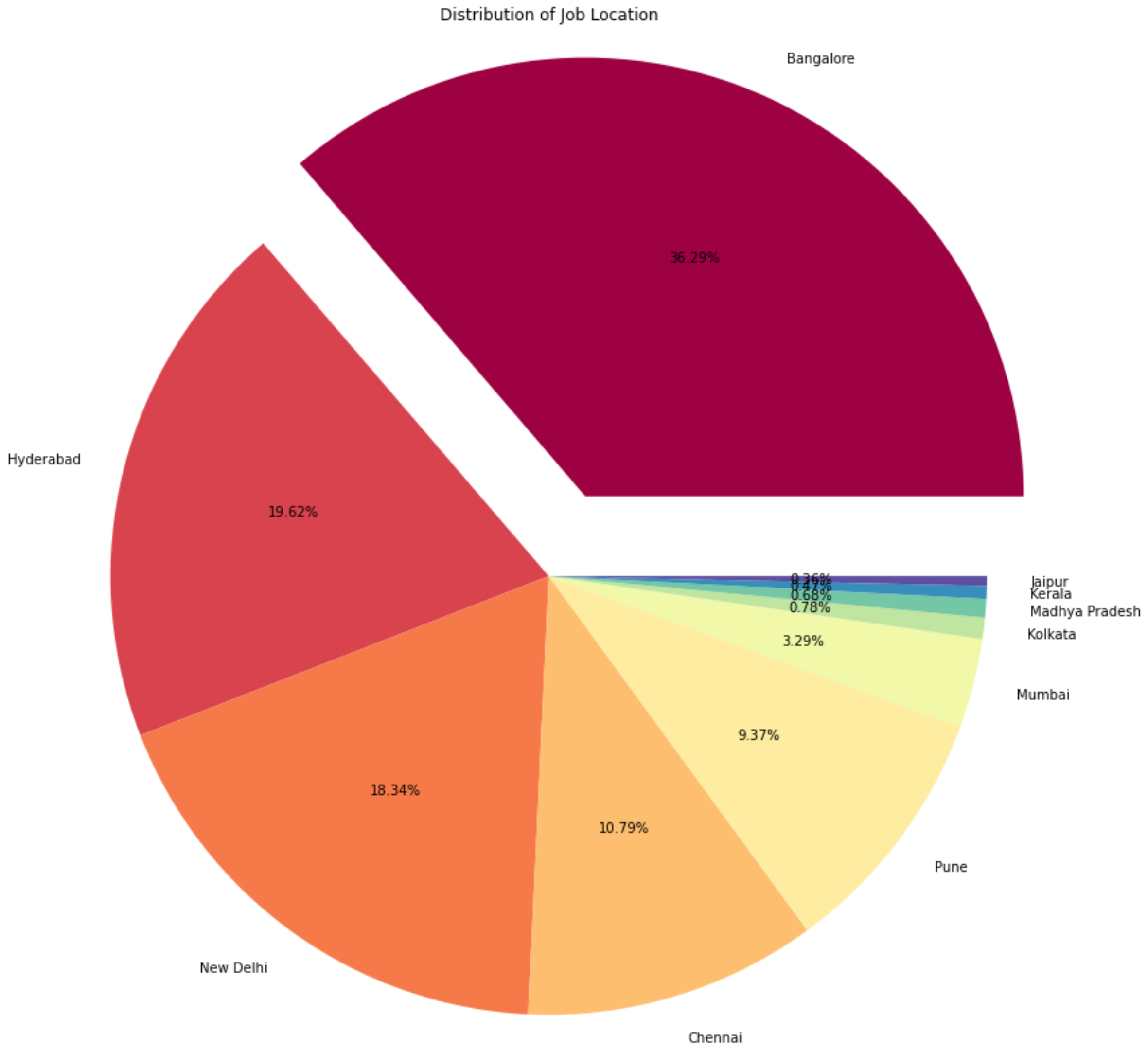
In [94]:

```
explode=(0.2,0,0,0)
df.sort_values("Salary", axis = 0, ascending = False, inplace = True)
df["Employment Status"].value_counts().plot.pie(figsize=(10,10),explode=explode,
ax = plt.gca()
ax.axes.yaxis.set_visible(False)
```



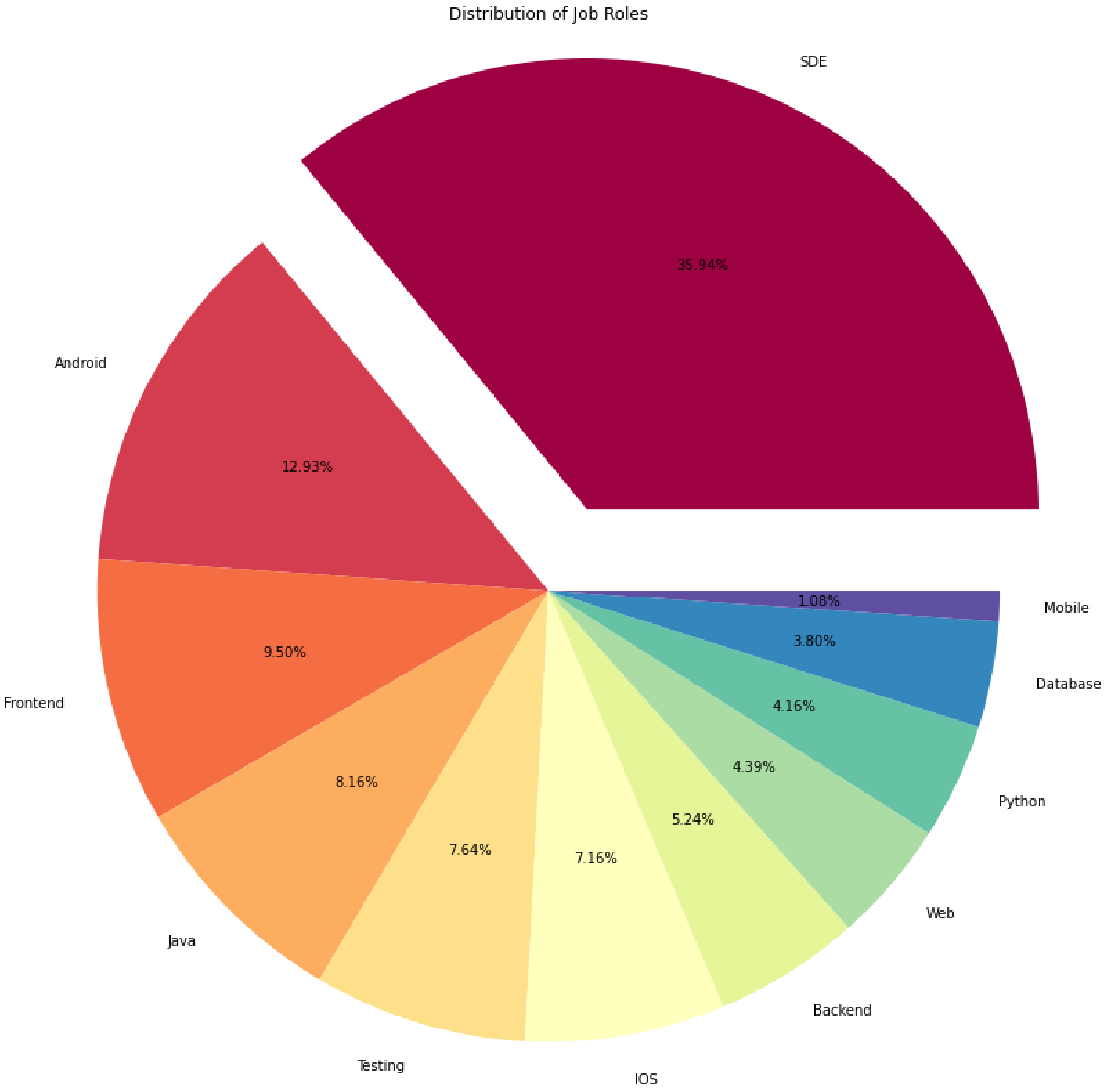
In [95]:

```
 0, 0, 0, 0, 0, 0, 0)
_counts().plot.pie(figsize = (20,15),explode=explode,autopct='%0.2f%%',colormap="Spec

;ible(False)
```



In [96]:

```
explode = (0.2,0,0,0,0,0,0,0,0,0,0,0)
df["Job Roles"].value_counts().plot.pie(figsize=(20,15),explode=explode,autopct='%0.
ax = plt.gca()
ax.axes.yaxis.set_visible(False)
```



Correlation

In [103]:

```
x = df.drop('Salary',axis=1) #Feature Matrix
y = df['Salary']
```

In [105]:

```
x.head()
```

Out[105]:

	Rating	Company Name	Job Title	Salaries Reported	Location	Employment Status	Job Roles
18635	3.6	Thapar University	Software Development Engineer (SDE)	1	New Delhi	Full Time	SDE
7121	3.5	Koru UX Design	Senior Front End Developer	1	Pune	Full Time	Frontend
9260	3.6	OASYS Cybernetics	Senior Java Developer	1	Chennai	Full Time	Java
4471	3.8	Concentrix	Oracle Database Administrator	1	Bangalore	Full Time	Database
5819	3.7	Nityo Infotech	Lead UI Designer, Magento Front-end Developer	1	Bangalore	Full Time	Frontend

In [107]:

```
# Separate database into train and test
from sklearn.model_selection import train_test_split
X_train,y_train,X_test,y_test = train_test_split(
x,
y,
test_size = 0.3,
random_state=0
)
X_train.shape, X_test.shape

```

Out[107]:

```
((15939, 7), (15939,))
```

In [111]:

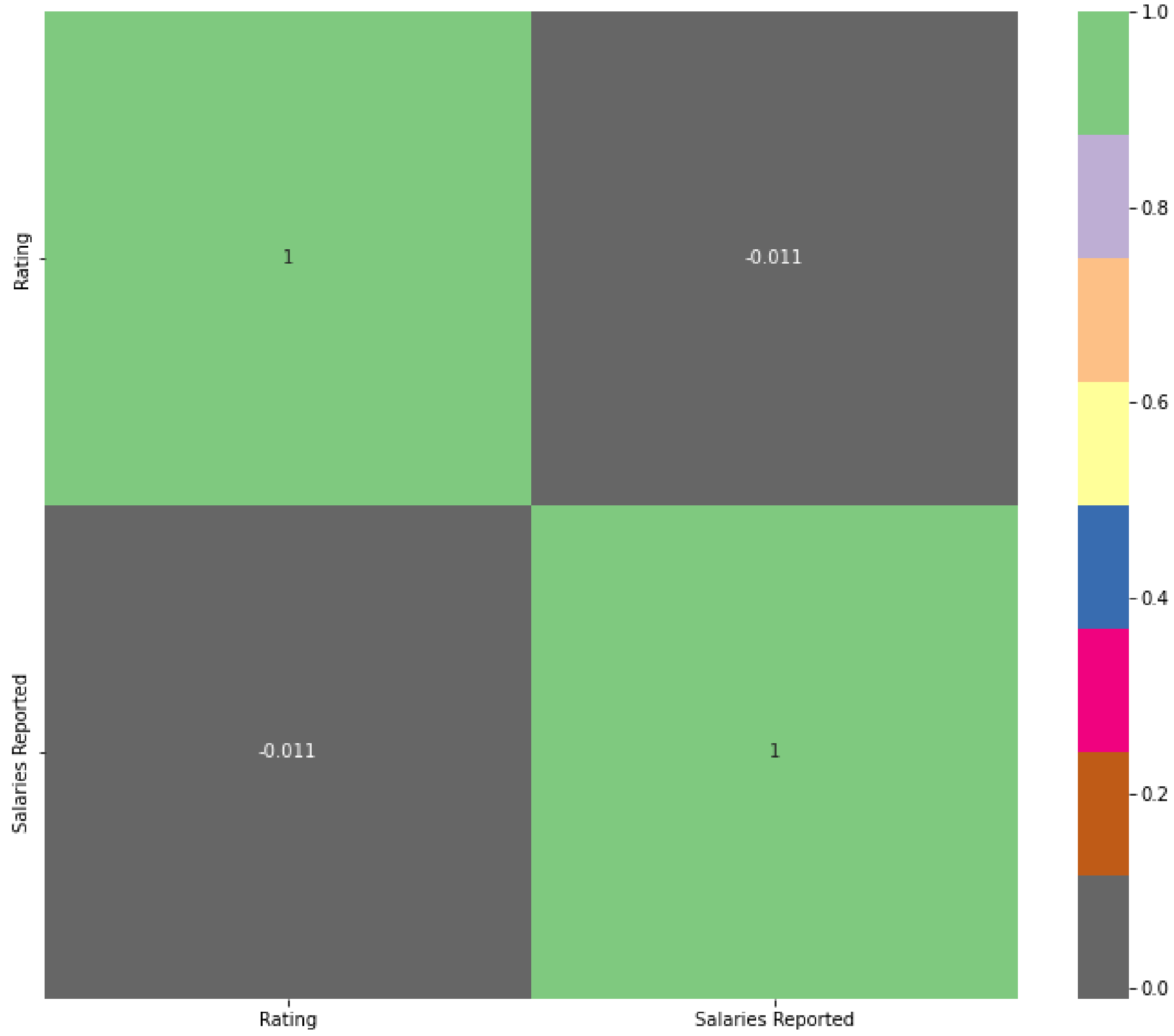
```
X_train.corr()
```

Out[111]:

	Rating	Salaries Reported
Rating	1.000000	-0.010635
Salaries Reported	-0.010635	1.000000

In [110]:

```
plt.figure(figsize=(12,10))
cor = X_train.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.Accent_r)
plt.show()
```



In [122]:

```
ollowing function we can select highly correlated features
move first feature that is correlated with anything other feature

ion(dataset, threshold):
 = set()
rix = dataset.corr()
 range(len(corr_matrix.columns)):
j in range(i):
if abs(corr_matrix.iloc[i,j])> threshold: #We are interested in absolute coeff value
   col_name = corr_matrix.columns[i]  # Getting the name of column
   col_corr.add(col_name)
ol_corr
```

In [127]:

```
corr_features = correlation(X_train,0.7)
len(set(corr_features))
```

Out[127]:

```
0
```

There is no no correlation in this dataset

Conclusion

1. SDE is the highest 'Job Role'

2. Software Development Engineer is the highest 'Job Title'

3. 'Bangalore' has the highest rate in Information Technology

4. There are big differences in salaries between cities for the same job role and same job title(We didn't see any data about the expertise of work, But we can put it into our consideration to understand these big differences in salaries

5. 'Full time' is the highest employeement status

6. Instead maximum people work as SDE but they can't pay much by companies. We can see Database job role payed more rather than other job roles.

7. There is no no correlation in this dataset(pearson coefficient=0)

In [ ]:

```

```