import require library

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

import the dataset

In [2]:
```
dataset = pd.read_csv(r'C:\Users\hp\OneDrive\Documents\Desktop\Salary_Data
```

Split the data to indipendent variable

In [3]:
```
x = dataset.iloc[:, :-1].values
```

In [6]:
```
dataset.head(1)
```

Out[6]:

	YearsExperience	Salary
0	1.1	39343

Split the data to dependent variable

In [4]:
```
y = dataset.iloc[:,1].values
```

In [7]:
```
dataset.head(1)
```

Out[7]:

	YearsExperience	Salary
0	1.1	39343

As dependent variable is continue that regression algorithm

As in the data set we have 2 attributes we slr algorithm

split the data to 80-20%

In [8]:
```
from sklearn.model_selection import train_test_split
```

In [10]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.20, ra
```

we called simple linear regression algoriytm from sklearm framework

In [11]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
regressor = LinearRegression()
```

we built simple linear regression model regressor

In [15]:
```
regressor.fit(x_train, y_train)
```

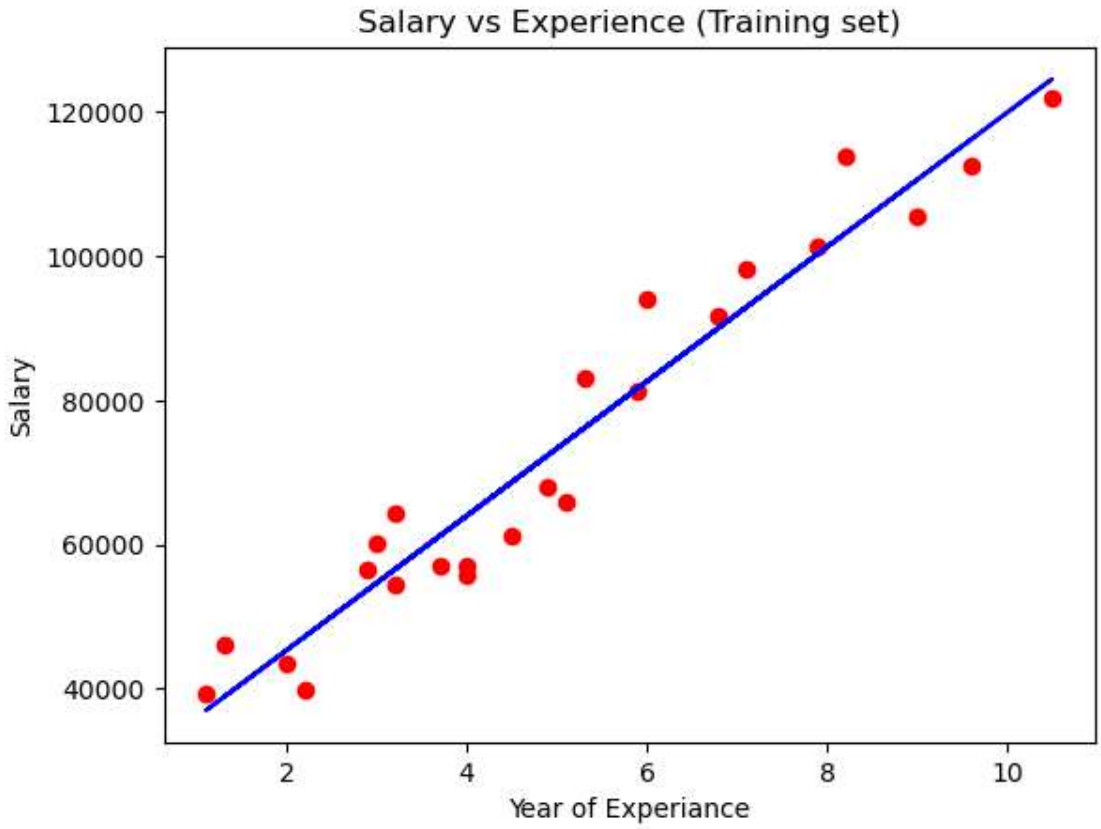
Out[15]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

test the model & create a predicted table

In [16]:
```
y_pred = regressor.predict(x_test)
```

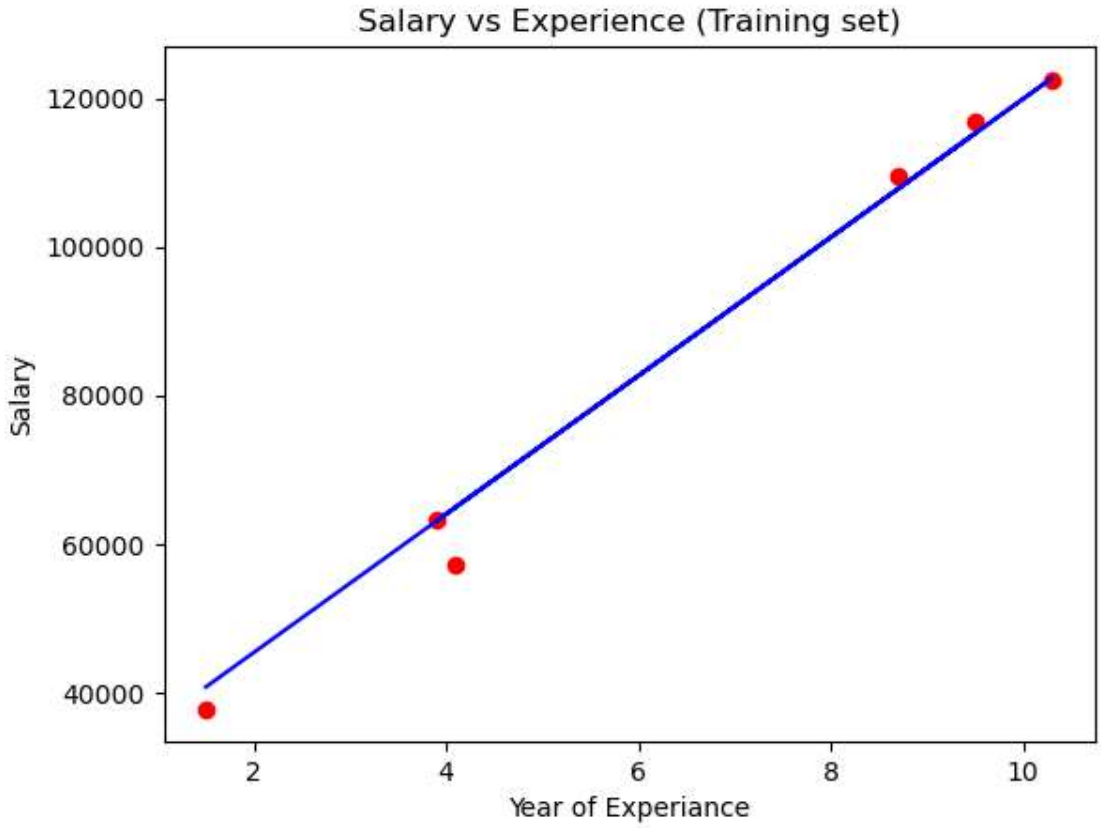
visualize train data point(24 data)

In [17]:
```
plt.scatter(x_train, y_train, color = "red")
plt.plot(x_train, regressor.predict(x_train), color = "blue")
plt.title("Salary vs Experience (Training set)")
plt.xlabel("Year of Experiance")
plt.ylabel("Salary")
plt.show()
```



visualize test data point

In [19]:
```
plt.scatter(x_test, y_test, color = "red")
plt.plot(x_test, regressor.predict(x_test), color = "blue")
plt.title("Salary vs Experience (Training set)")
plt.xlabel("Year of Experiance")
plt.ylabel("Salary")
plt.show()
```



slope is generated from linear regression algorith which fit

In [21]:
```
m = regressor.intercept_
```

intercept also generated by the model

In [23]:
```
c = regressor.intercept_
```

predict or forcast the future the data which we not trainebefore

In [25]:
```
y_12 = 9312 * 12 + 26780
y_12
```

Out[25]: 138524

In [28]:
```
y_20 = 9312 * 12 + 267#80
y_20
```

Out[28]: 112011

to check overfitting (low bias high variance)

In [29]:
```
bias = regressor.score(x_train, y_train)
bias
```

Out[29]: 0.9411949620562126

In [30]:
```
# to check underfitting (high bias low variance)
```

In [31]:
```
variance = regressor.score(x_test,y_test)
variance
```

Out[31]: 0.988169515729126

In [ ]:
```

```