Praticle 3 : Implementation of Random Forest Regression

In [42]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
```

In [43]:
```
# Load the dataset
data = pd.read_csv('salary_data.csv')
```

In [44]:
```
# Handling missing values in the feature variable
imputer = SimpleImputer(strategy='mean')
data['Years of Experience'] = imputer.fit_transform(data[['Years of Exper

# Handling missing values in the target variable
data['Salary'] = imputer.fit_transform(data[['Salary']])
```

In [45]:
```
# Splitting the data into features (X) and the target variable (y)
X = data[['Years of Experience']]
y = data['Salary']
```

In [46]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

In [47]:
```
# Train the Random Forest model
regressor = RandomForestRegressor(n_estimators=100, random_state=42)
regressor.fit(X_train, y_train)
```

Out[47]:
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [48]:
```
# Predict on the test set
y_pred = regressor.predict(X_test)
```

In [49]:
```
print("Test Data:")
print(X_test)
print("Actual Salary:")
print(y_test)
```

```
Test Data:
     Years of Experience
167                 18.0
33                  10.0
15                  16.0
316                  6.0
57                  17.0
..                   ...
94                   7.0
196                 11.0
350                 16.0
312                 15.0
349                  8.0

[75 rows x 1 columns]
Actual Salary:
167    150000.0
33      65000.0
15     125000.0
316     80000.0
57     140000.0
         ...
94      75000.0
196     90000.0
350    160000.0
312    150000.0
349    110000.0
Name: Salary, Length: 75, dtype: float64
```

In [50]:
```
print("Predicted Salary:")
print(y_pred)
```

```
Predicted Salary:
[150030.86858411  98177.09723669 157949.80830386  82967.23706849
 145649.41738817 174561.75840499 162394.08924409  98177.09723669
  51269.10846103 102283.17040389 119067.50556934 174561.75840499
  40507.77998257 102283.17040389  59138.35244112 157949.80830386
  60807.06685407  34939.14402289  84463.79852453  86800.93917322
  98177.09723669 102283.17040389 157949.80830386  60807.06685407
  98177.09723669  59138.35244112 166373.93105196  99608.10145023
 150030.86858411 167255.32323778  51269.10846103 129451.00281663
  51269.10846103 133078.7445125   84463.79852453  35429.66450216
 174561.75840499  40507.77998257  40507.77998257  59138.35244112
  40507.77998257  86800.93917322 197230.         167255.32323778
  40507.77998257 157949.80830386  98577.40731491  40507.77998257
  40507.77998257  40507.77998257  98177.09723669  86800.93917322
  86800.93917322 104986.41276669 166373.93105196  59138.35244112
 197230.          84463.79852453  51269.10846103  82967.23706849
 104986.41276669 129451.00281663 166373.93105196  60807.06685407
  40507.77998257  40507.77998257  51269.10846103 129451.00281663
  40507.77998257  35429.66450216  84463.79852453  98577.40731491
 157949.80830386 133078.7445125   86800.93917322]
```

In [51]:
```
# Visualization of testing data
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.scatter(X_test, y_pred, color='red', label='Predicted')
plt.title('Random Forest Regression')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
```

```
plt.legend()
plt.show()
```