In [2]:

```
import pandas as pd
import numpy as np
```

In [64]:

```
file_path = "C:/Users/preet/Downloads/myexcel - myexcel.csv.csv"
data = pd.read_csv(file_path)

print(data.head())
```

```
            Name            Team  Number Position  Age  Height  Weight  \
0  Avery Bradley  Boston Celtics       0       PG   25  06-Feb     180   
1    Jae Crowder  Boston Celtics      99       SF   25  06-Jun     235   
2   John Holland  Boston Celtics      30       SG   27  06-May     205   
3    R.J. Hunter  Boston Celtics      28       SG   22  06-May     185   
4  Jonas Jerebko  Boston Celtics       8       PF   29  06-Oct     231   

             College     Salary  
0              Texas  7730337.0  
1          Marquette  6796117.0  
2  Boston University        NaN  
3      Georgia State  1148640.0  
4                NaN  5000000.0  
```

In [8]:

```
if "height" in data.columns:

    data["height"] = np.random.randint(150, 181, size=len(data))
    print("Height column updated successfully.")
else:
    print("The 'height' column is missing in the dataset.")
```

```
The 'height' column is missing in the dataset.
```

In [12]:

```
if "team" in data.columns:
    team_distribution = data["team"].value_counts()
    team_percentage = (team_distribution / len(data)) * 100


    team_summary = pd.DataFrame({
        "Employee Count": team_distribution,
        "Percentage": team_percentage})

    print("Team Distribution:")
    print(team_summary)
else:
    print("The 'team' column is missing in the dataset.")
```

```
The 'team' column is missing in the dataset.
```

In [16]:

```
if "position" in data.columns:
    position_distribution = data["position"].value_counts()


    position_summary = pd.DataFrame({ "Employee Count": position_distribution})

    print("\nPosition Distribution:")
    print(position_summary)
```

```
else:
    print("The 'position' column is missing in the dataset.")
```

```
The 'position' column is missing in the dataset.
```

In [20]:

```
if "age" in data.columns:

    bins = [0, 20, 30, 40, 50, 60, 100]
    labels = ["<20", "20-30", "30-40", "40-50", "50-60", "60+"]
    data["age_group"] = pd.cut(data["age"], bins=bins, labels=labels, right=False)


    age_group_distribution = data["age_group"].value_counts()

    print("Age Group Distribution:")
    print(age_group_distribution)
else:
    print("The 'age' column is missing in the dataset.")
```

```
The 'age' column is missing in the dataset.
```

In [24]:

```
if all(col in data.columns for col in ["team", "position", "salary"]):

    salary_summary = data.groupby(["team", "position"])["salary"].sum()


    max_salary_expenditure = salary_summary.idxmax()
    max_salary_value = salary_summary.max()

    print(Team and Position with the Highest Salary Expenditure: {max_salary_expend
    print(Total Salary Expenditure: {max_salary_value}")
else:
    print("One or more columns ('team', 'position', 'salary') are missing in the da
```

```
One or more columns ('team', 'position', 'salary') are missing in the dataset.
```

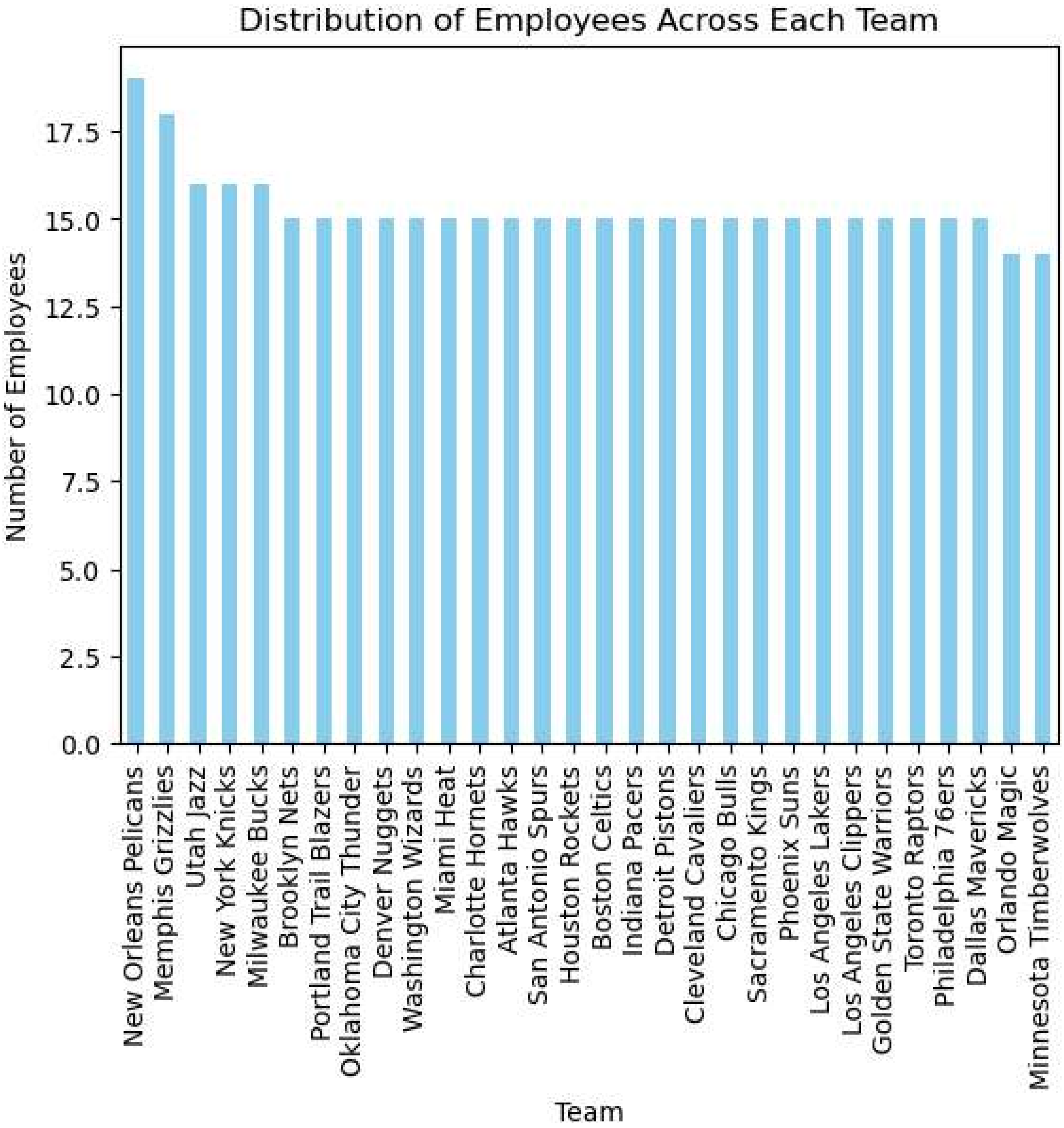
In [52]:

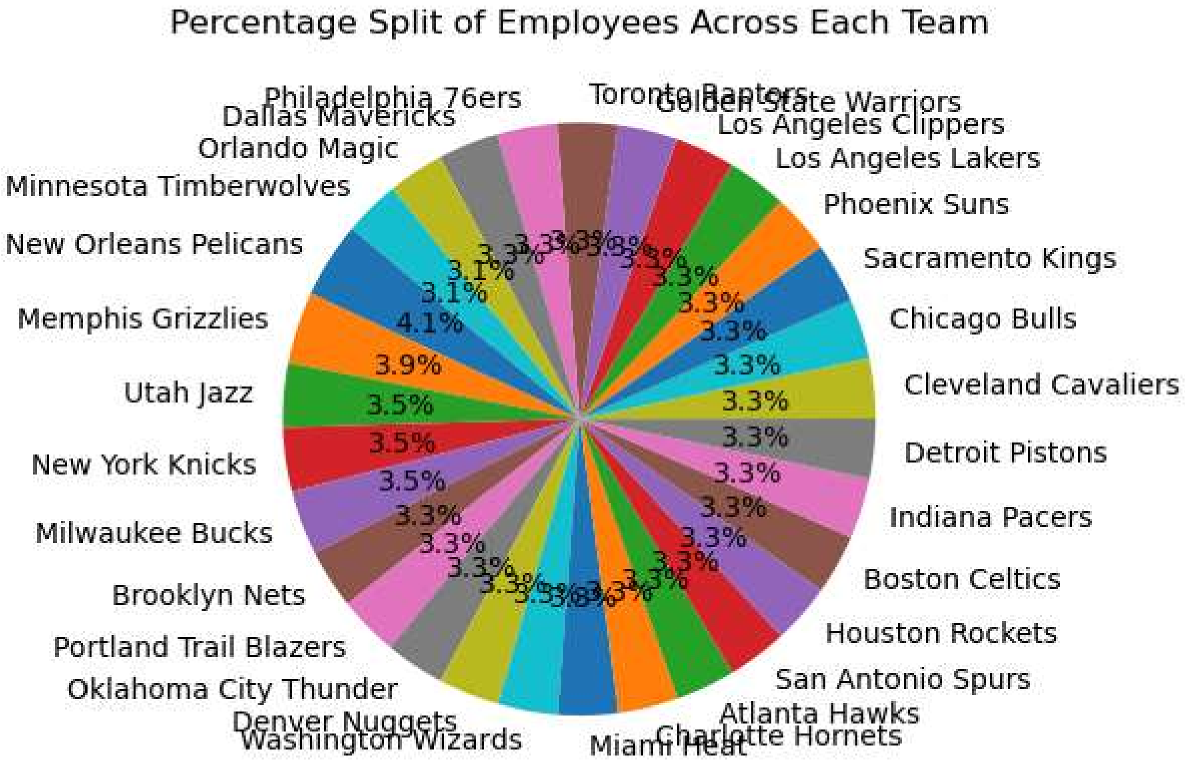
```
import pandas as pd

file_path = "C:/Users/preet/Downloads/myexcel - myexcel.csv.csv"
data = pd.read_csv(file_path)
print(data.columns)
```

```
Index(['Name', 'Team', 'Number', 'Position', 'Age', 'Height', 'Weight',
       'College', 'Salary'],
      dtype='object')
```

In [54]:

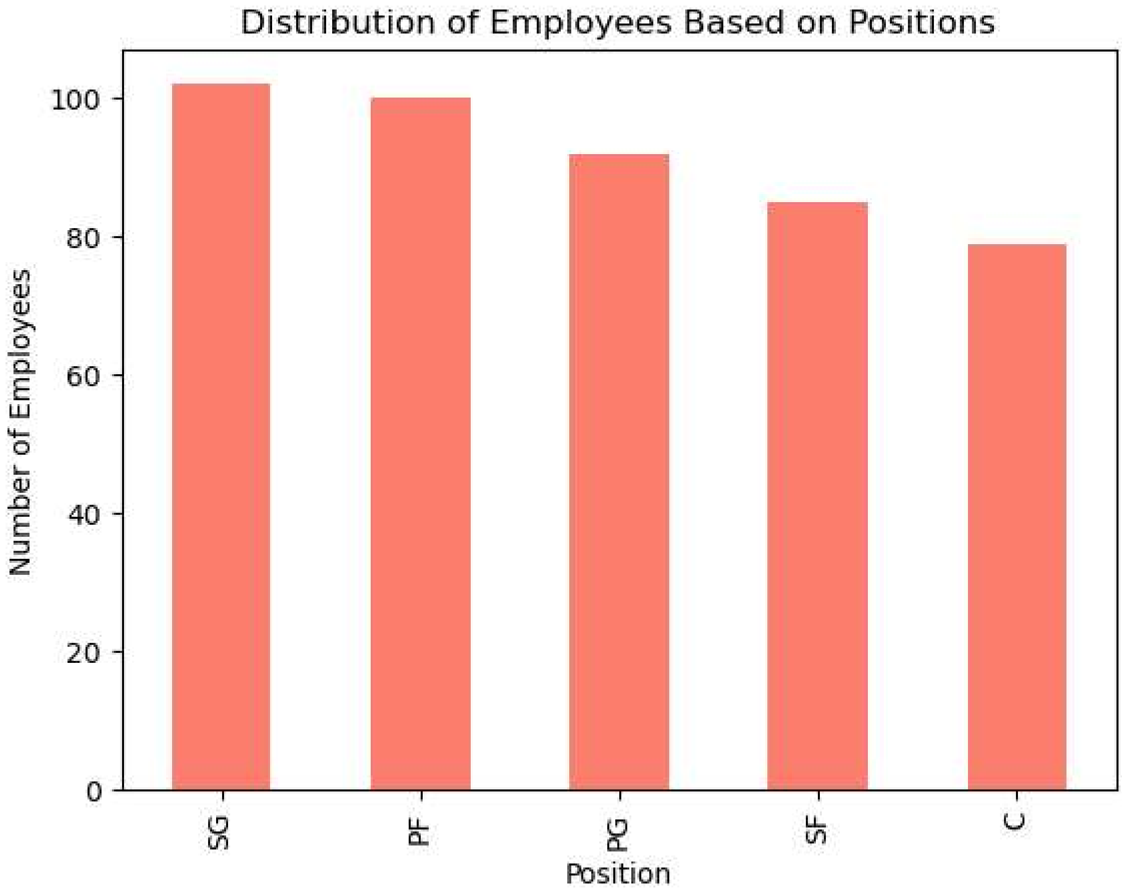
```
import pandas as pd
import matplotlib.pyplot as plt

file_path = "C:/Users/preet/Downloads/myexcel - myexcel.csv.csv"
data = pd.read_csv(file_path)

team_distribution = data['Team'].value_counts()

team_distribution.plot(kind='bar', color='skyblue')
plt.title('Distribution of Employees Across Each Team')
plt.xlabel('Team')
plt.ylabel('Number of Employees')
plt.show()

total_employees = len(data)
percentage_split = (team_distribution / total_employees) * 100
percentage_split.plot(kind='pie', startangle=140)
plt.title('Percentage Split of Employees Across Each Team')
plt.ylabel('')
plt.show()
```





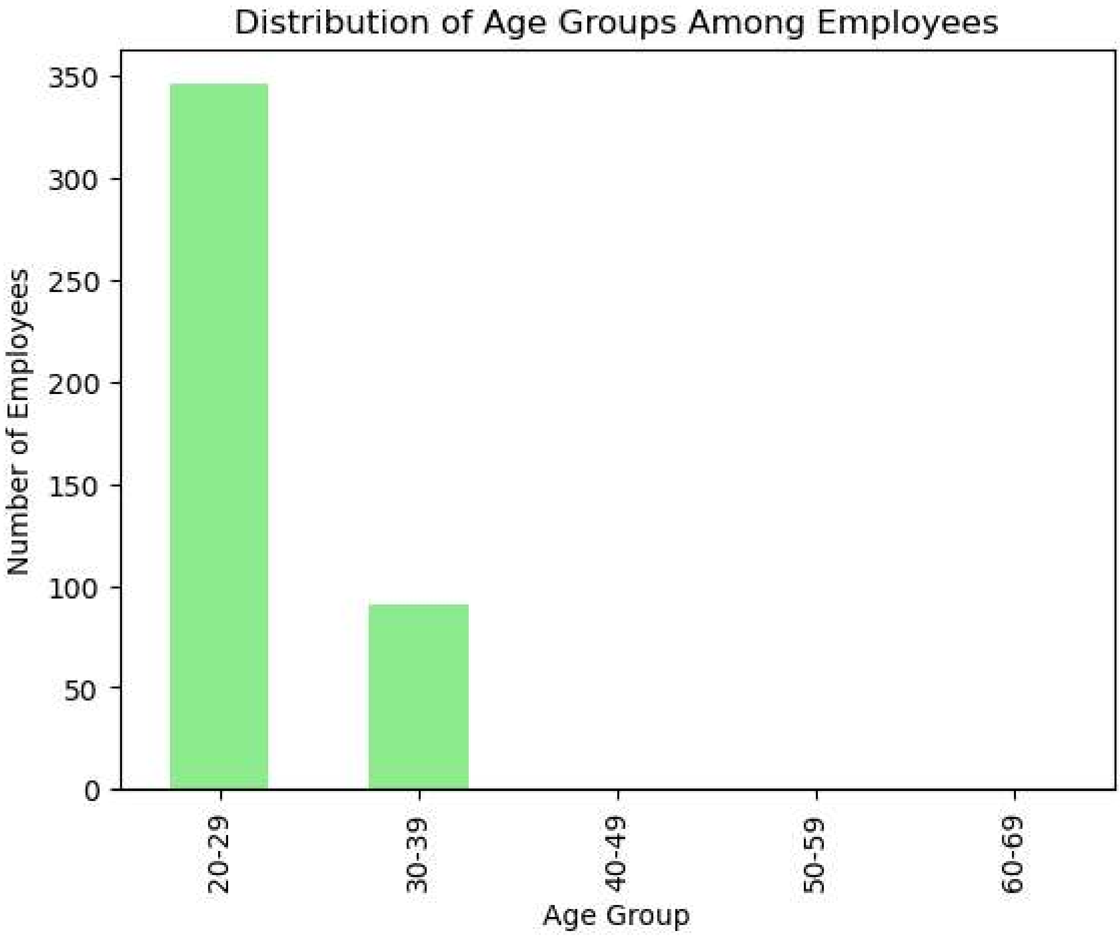
In [56]:

```
position_distribution = data['Position'].value_counts()
position_distribution.plot(kind='bar', color='salmon')
plt.title('Distribution of Employees Based on Positions')
plt.xlabel('Position')
plt.ylabel('Number of Employees')
plt.show()
```



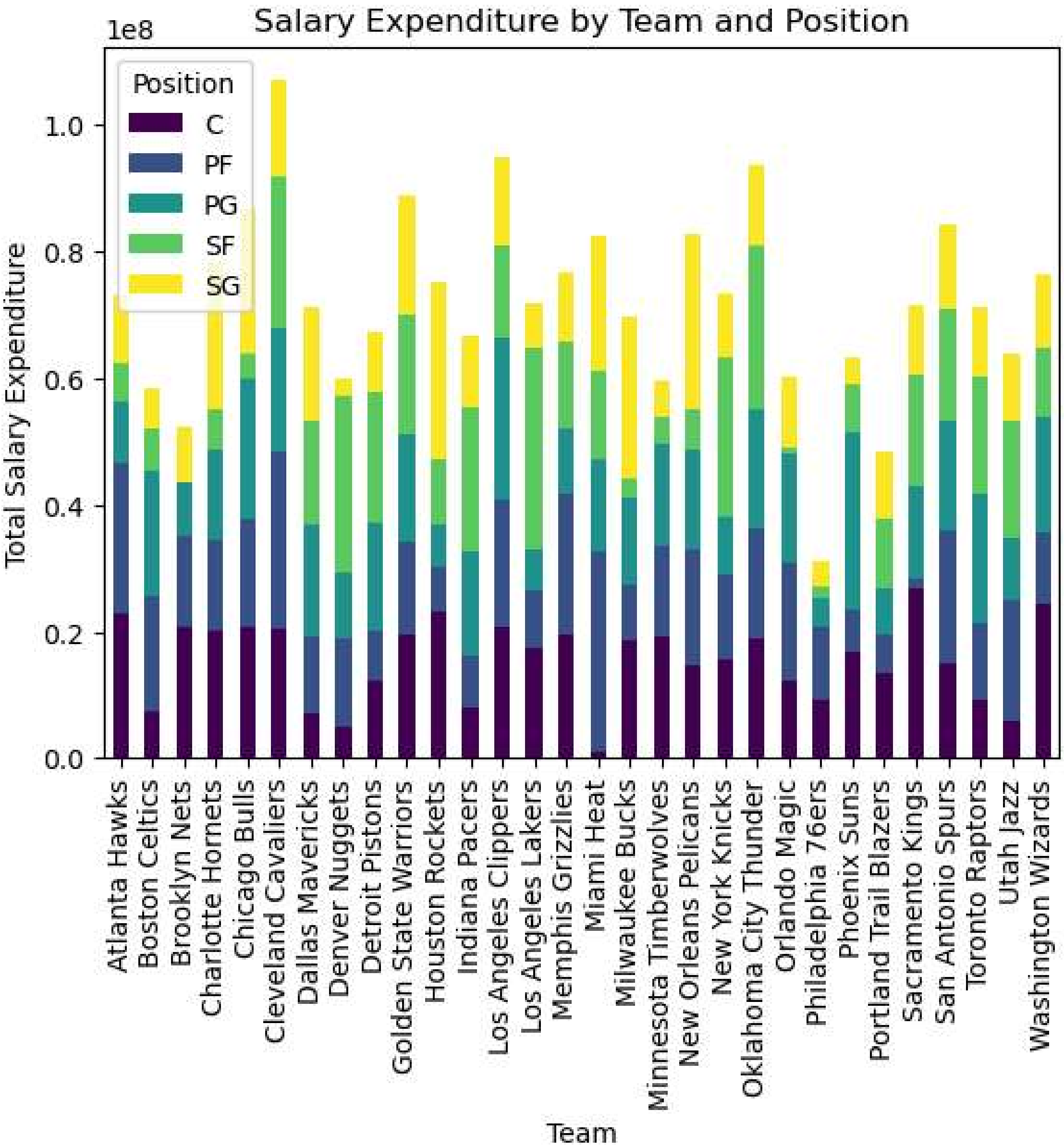
In [58]:

```python
bins = [20, 30, 40, 50, 60, 70]
labels = ['20-29', '30-39', '40-49', '50-59', '60-69']

data['age_group'] = pd.cut(data['Age'], bins=bins, labels=labels)

age_group_distribution = data['age_group'].value_counts()

age_group_distribution.plot(kind='bar', color='lightgreen')
plt.title('Distribution of Age Groups Among Employees')
plt.xlabel('Age Group')
plt.ylabel('Number of Employees')
plt.show()
```



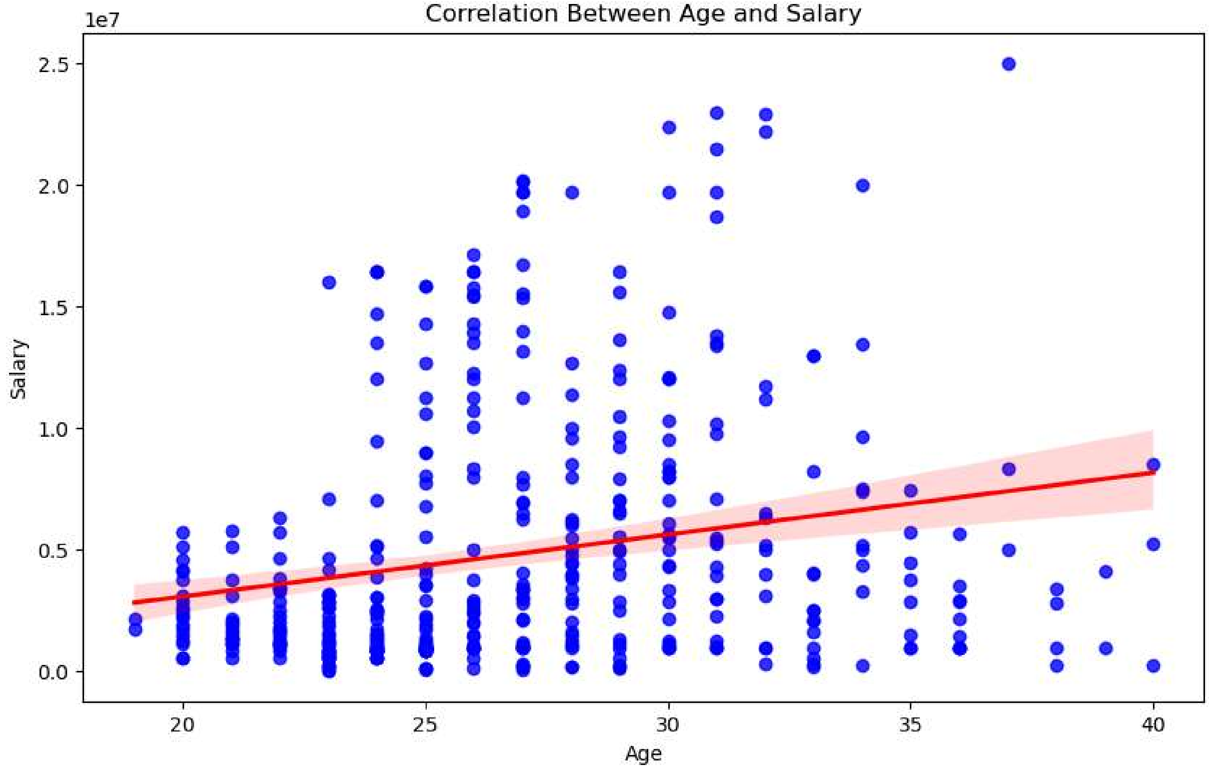
In [60]:

```
salary_expenditure = data.groupby(['Team', 'Position'])['Salary'].sum()

salary_expenditure.unstack().plot(kind='bar', stacked=True, colormap='viridis')
plt.title('Salary Expenditure by Team and Position')
plt.xlabel('Team')
plt.ylabel('Total Salary Expenditure')
plt.legend(title='Position')
plt.show()
```



In [62]:

```
import seaborn as sns

correlation = data['Age'].corr(data['Salary'])

plt.figure(figsize=(10, 6))
sns.regplot(x='Age', y='Salary', data=data, scatter_kws={"color": "blue"}, line_kws
plt.title('Correlation Between Age and Salary')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()

print("Correlation between age and salary: {correlation}")
```



```
Correlation between age and salary: 0.21400941226570974
```

In [ ]:

```
Data Story:
Key Insights:

Distribution of Employees Across Each Team:
Insight: The bar and pie charts reveal the distribution of employees across differe

Segregation of Employees Based on Their Positions:
Insight: The bar chart shows the number of employees in each position, highlighting

Predominant Age Group Among Employees:
Insight: The bar chart indicates the most represented age group, providing demograp

Team and Position with the Highest Salary Expenditure:
Insight: The stacked bar chart shows the salary expenditure by team and position, i

Correlation Between Age and Salary:
Insight: The scatter plot with a regression line visualizes the relationship betwee
```