EDA

Which month had the highest Forcasted calls?

In [9]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_excel(r"C:\Users\10\OneDrive\Desktop\تاسك\Salla\Salla Call Center Dataset - NEW.xlsx")
```

In [11]:

```
df
```

Out[11]:

	Project	Date	Month	Forecasted Calls	Calls Offered	Calls Handled	Calls Handled With in Thrshold	Calls Abandon	ASA	Answer Time
0	Project A	2022-02-01	Feb-2022	13473	8393	8381	8380	12	1.689178	14157.0
1	Project A	2022-02-02	Feb-2022	13255	5144	5134	5096	10	2.337164	11999.0
2	Project A	2022-02-03	Feb-2022	13209	8450	8436	8432	14	1.874822	15816.0
3	Project A	2022-02-04	Feb-2022	13002	10883	10614	7376	269	51.207744	543519.0
4	Project A	2022-02-05	Feb-2022	12758	11445	11393	11027	52	7.235232	82431.0
...	...	...	...	...	...	...	...	...	...	...
262	Project C	2022-04-26	Apr-2022	294	164	163	163	1	1.883436	307.0
263	Project C	2022-04-27	Apr-2022	291	166	158	145	8	8.411392	1329.0
264	Project C	2022-04-28	Apr-2022	288	149	142	130	7	7.429577	1055.0
265	Project C	2022-04-29	Apr-2022	219	137	136	132	1	4.426471	602.0
266	Project C	2022-04-30	Apr-2022	259	100	97	89	3	8.628866	837.0

267 rows × 10 columns

In [13]:

```
print(df.columns)
print(df.shape)
```

```
Index(['Project', 'Date', 'Month', 'Forecasted Calls', 'Calls Offered',
       'Calls Handled', 'Calls Handled With in Thrshold', 'Calls Abandon',
       'ASA', 'Answer Time'],
      dtype='object')
(267, 10)
```

In [6]:

```
df['Date']=df['Date'].astype(str).str.replace('2022','2024')
```

In [15]:

```
df
```

Out[15]:

	Project	Date	Month	Forecasted Calls	Calls Offered	Calls Handled	Calls Handled With in Thrshold	Calls Abandon	ASA	Answer Time
0	Project A	2022-02-01	Feb-2022	13473	8393	8381	8380	12	1.689178	14157.0
1	Project A	2022-02-02	Feb-2022	13255	5144	5134	5096	10	2.337164	11999.0
2	Project A	2022-02-03	Feb-2022	13209	8450	8436	8432	14	1.874822	15816.0
3	Project A	2022-02-04	Feb-2022	13002	10883	10614	7376	269	51.207744	543519.0
4	Project A	2022-02-05	Feb-2022	12758	11445	11393	11027	52	7.235232	82431.0
...	...	...	...	...	...	...	...	...	...	...
262	Project C	2022-04-26	Apr-2022	294	164	163	163	1	1.883436	307.0
263	Project C	2022-04-27	Apr-2022	291	166	158	145	8	8.411392	1329.0
264	Project C	2022-04-28	Apr-2022	288	149	142	130	7	7.429577	1055.0
265	Project C	2022-04-29	Apr-2022	219	137	136	132	1	4.426471	602.0
266	Project C	2022-04-30	Apr-2022	259	100	97	89	3	8.628866	837.0

267 rows × 10 columns

In [55]:

```
df['Month']= df['Month'].astype(str).str.replace('2022', '2024')
```

In [132]:

```
import matplotlib.pyplot as plt
calls_handled_by_month=df.groupby('Month')['Calls Handled'].sum()
calls_handled_by_month.plot(kind='bar',color='blue')
plt.xlabel('Month')
plt.ylabel('Calls Handled')
plt.xticks(rotation=45, ha='right')
plt.title('Total Calls Handled by Month')
```

Out[132]:

```
Text(0.5, 1.0, 'Total Calls Handled by Month')
```

In [17]:

```
df
```

Out[17]:

	Project	Date	Month	Forecasted Calls	Calls Offered	Calls Handled	Calls Handled With in Thrshold	Calls Abandon	ASA	Answer Time
0	Project A	2022-02-01	Feb-2022	13473	8393	8381	8380	12	1.689178	14157.0
1	Project A	2022-02-02	Feb-2022	13255	5144	5134	5096	10	2.337164	11999.0
2	Project A	2022-02-03	Feb-2022	13209	8450	8436	8432	14	1.874822	15816.0
3	Project A	2022-02-04	Feb-2022	13002	10883	10614	7376	269	51.207744	543519.0
4	Project A	2022-02-05	Feb-2022	12758	11445	11393	11027	52	7.235232	82431.0
...	...	...	...	...	...	...	...	...	...	...
262	Project C	2022-04-26	Apr-2022	294	164	163	163	1	1.883436	307.0
263	Project C	2022-04-27	Apr-2022	291	166	158	145	8	8.411392	1329.0
264	Project C	2022-04-28	Apr-2022	288	149	142	130	7	7.429577	1055.0
265	Project C	2022-04-29	Apr-2022	219	137	136	132	1	4.426471	602.0
266	Project C	2022-04-30	Apr-2022	259	100	97	89	3	8.628866	837.0

267 rows × 10 columns

In [19]:

```
df.describe(include='all')
```

Out[19]:

	Project	Date	Month	Forecasted Calls	Calls Offered	Calls Handled	Calls Handled With in Thrshold	Calls Abandon	ASA
count	267	267	267	267.000000	267.000000	267.000000	267.000000	267.000000	267.000000
unique	3	NaN	3	NaN	NaN	NaN	NaN	NaN	NaN
top	Project A	NaN	Mar-2022	NaN	NaN	NaN	NaN	NaN	NaN
freq	89	NaN	93	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	2022-03-17 00:00:00	NaN	9127.606742	6535.149813	6448.846442	5950.876404	86.303371	9.220929
min	NaN	2022-02-01 00:00:00	NaN	213.000000	81.000000	79.000000	75.000000	0.000000	1.578571
25%	NaN	2022-02-23 00:00:00	NaN	2674.000000	1834.500000	1824.500000	1804.000000	5.000000	1.875656
50%	NaN	2022-03-17 00:00:00	NaN	12477.000000	6601.000000	6592.000000	6291.000000	12.000000	2.743916
75%	NaN	2022-04-08 00:00:00	NaN	13208.000000	8885.000000	8809.000000	8153.500000	42.000000	6.748759
max	NaN	2022-04-30 00:00:00	NaN	17685.000000	30381.000000	29630.000000	22741.000000	3101.000000	180.903391
std	NaN	NaN	NaN	5926.564904	5255.978390	5118.075138	4571.721375	287.741271	19.585007

In [130]:

```
calls_forecasted_by_month=df.groupby('Month')['Forecasted Calls'].sum()
calls_forecasted_by_month.plot(kind='bar',color='blue')
plt.xlabel('Month')
plt.ylabel('Forecasted Calls')
plt.xticks(rotation=45, ha='right')
plt.title('Calls Forecasted By Month')
```

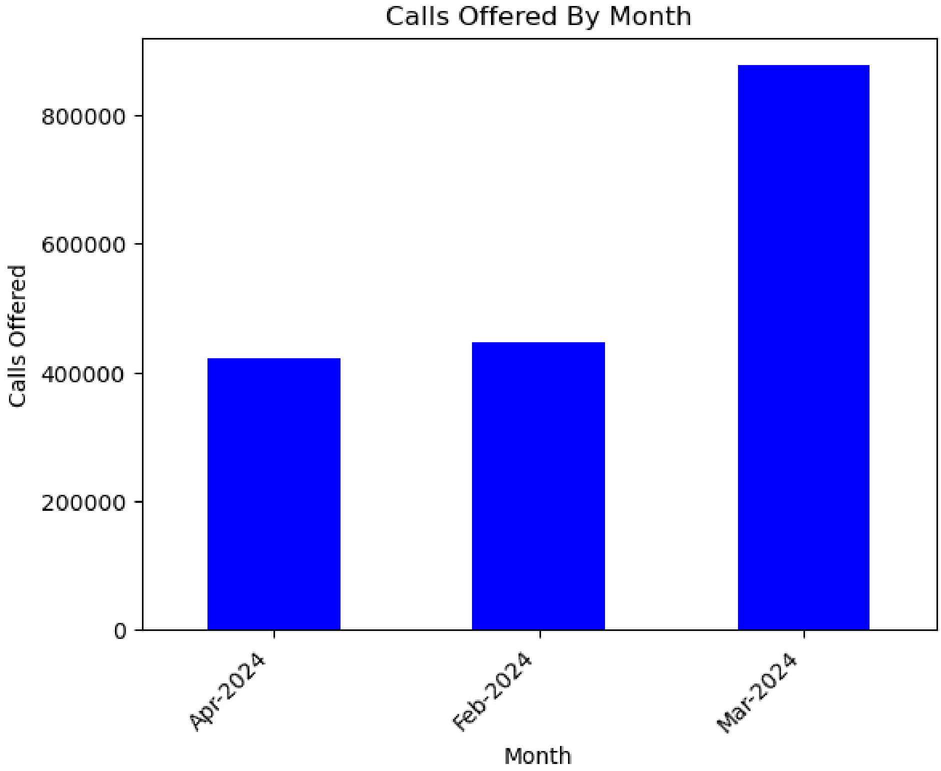
Out[130]:

```
Text(0.5, 1.0, 'Calls Forecasted By Month')
```

ماأكثر الشهور استقبالا للمكالمات؟

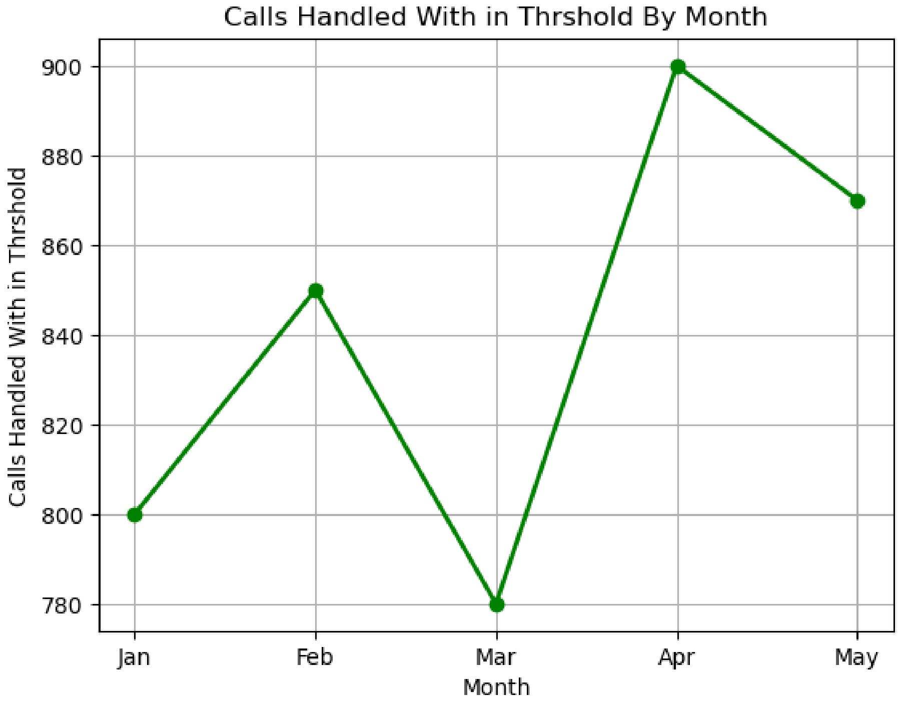
In [128]:

```
calls_offered_by_month=df.groupby('Month')['Calls Offered'].sum()
calls_offered_by_month.plot(kind='bar',color='blue')
plt.xlabel('Month')
plt.ylabel('Calls Offered')
plt.xticks(rotation=45, ha='right')
plt.title('Calls Offered By Month')
plt.show()
```



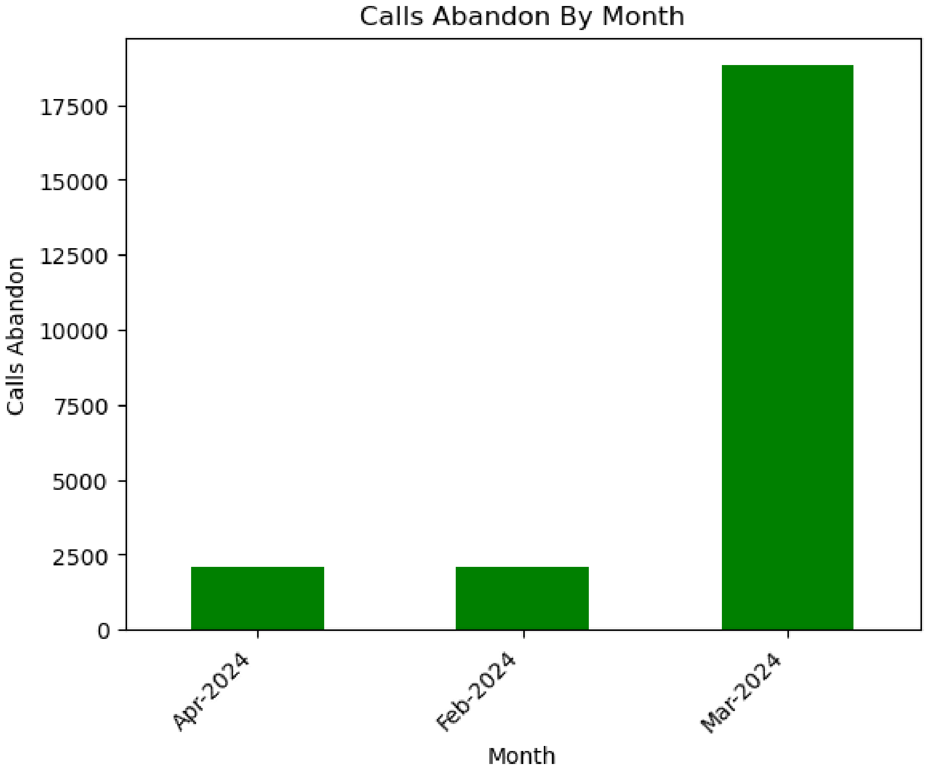
In [62]:

```
plt.plot(df['Month'], df['Calls Handled Within Threshold'], marker='o', linestyle='-', color='green', linewidth=2)
plt.xlabel('Month')
plt.ylabel('Calls Handled With in Thrshold')
plt.title('Calls Handled With in Thrshold By Month')
plt.grid(True)
plt.show()
```



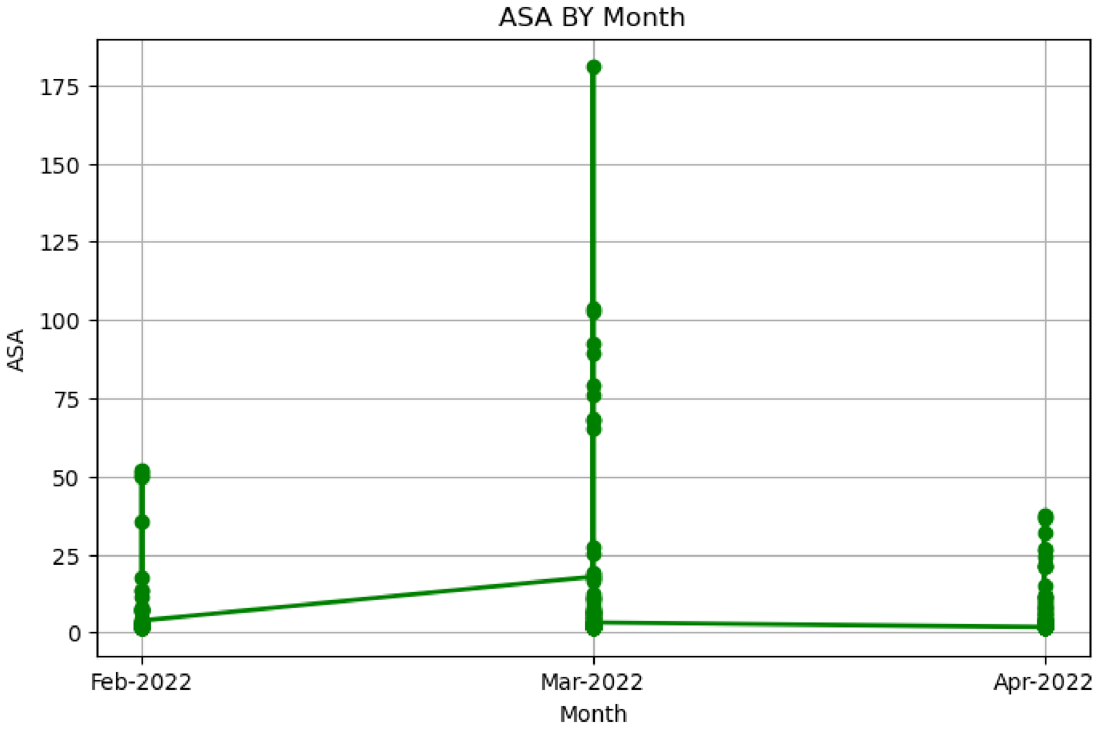
In [126]:

```
calls_abandon_by_month=df.groupby('Month')['Calls Abandon'].sum()
calls_abandon_by_month.plot(kind='bar',color='green')
plt.xlabel('Month')
plt.ylabel('Calls Abandon')
plt.title('Calls Abandon By Month')
plt.xticks(rotation=45, ha='right')
plt.show()
```



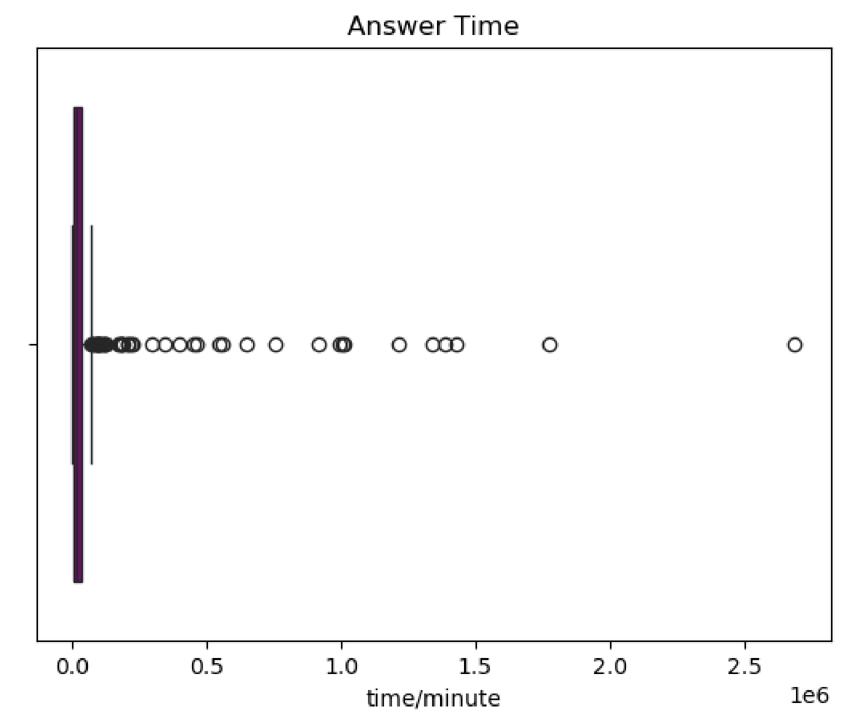
In [86]:

```
plt.figure(figsize=(8, 5))
plt.plot(df['Month'], df['ASA'], marker='o', linestyle='-', color='green', linewidth=2)

plt.xlabel('Month')
plt.ylabel('ASA')
plt.title('ASA BY Month')
plt.grid(True)
plt.show()
```



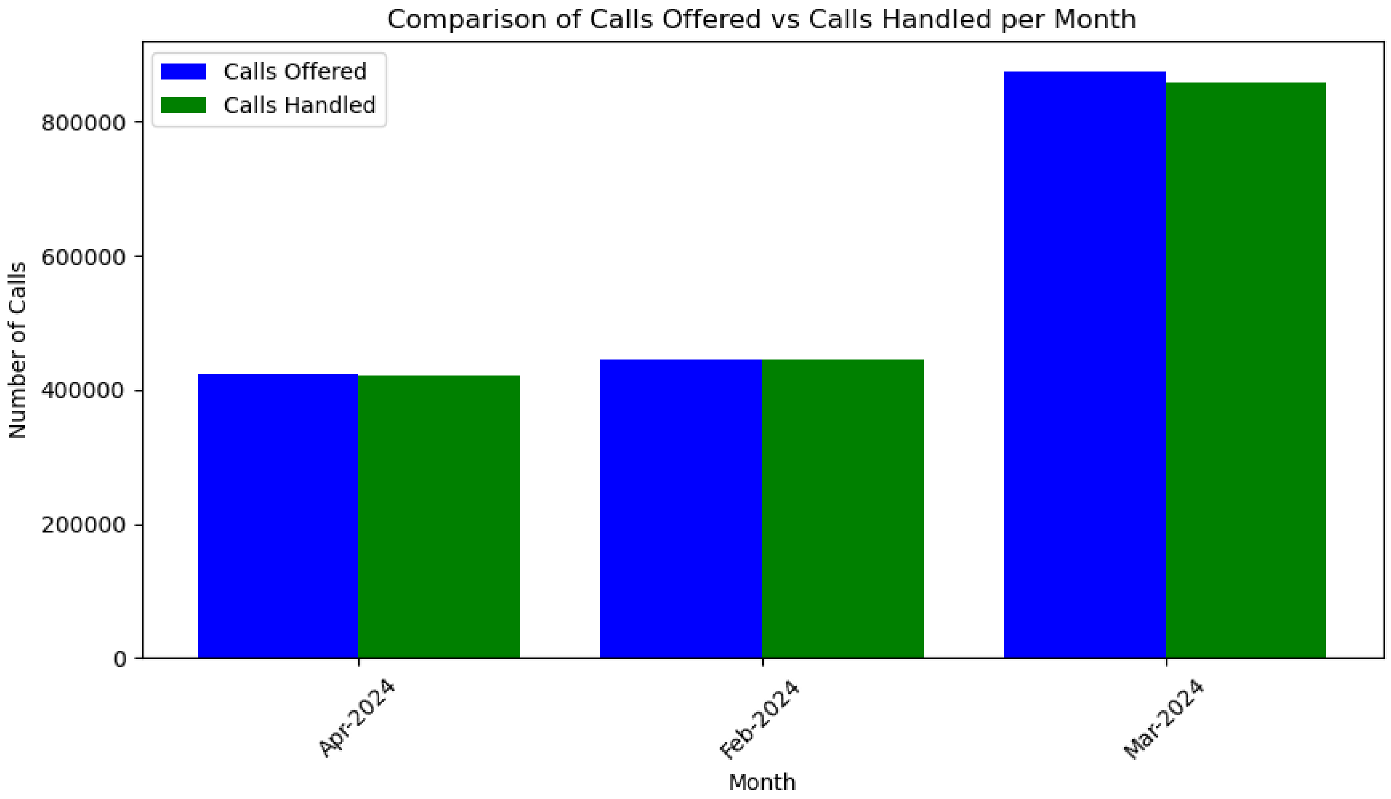
In [112]:

```
sns.boxplot(x=df['Answer Time'],
            color='purple')

plt.title('Answer Time')
plt.xlabel('time/minute')
plt.show()
```



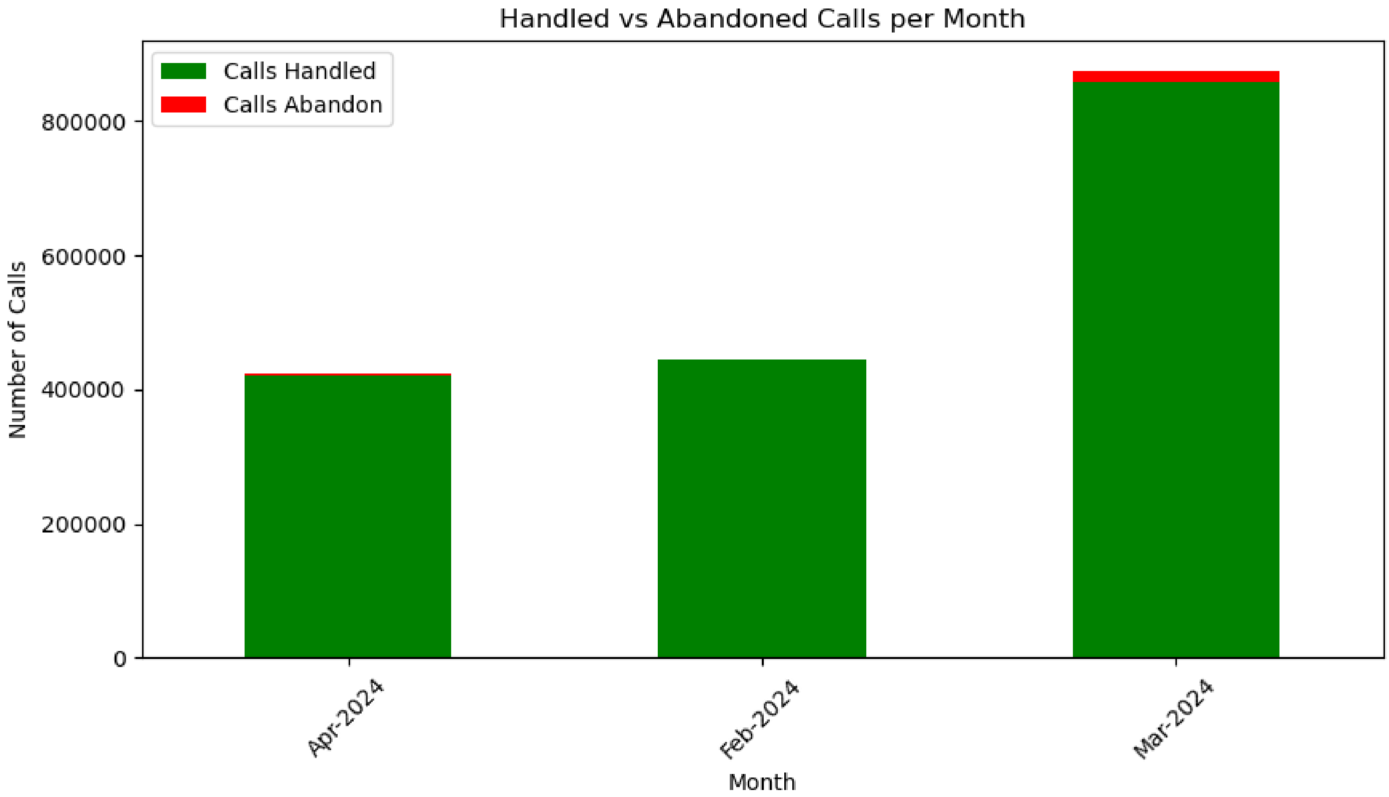
In [118]:

```
calls_data = df.groupby('Month')[['Calls Offered', 'Calls Handled']].sum()
months = calls_data.index
x = np.arange(len(months))
width = 0.4

fig, ax = plt.subplots(figsize=(10, 5))
ax.bar(x - width/2, calls_data['Calls Offered'], width, label='Calls Offered', color='blue')
ax.bar(x + width/2, calls_data['Calls Handled'], width, label='Calls Handled', color='green')

ax.set_xlabel('Month')
ax.set_ylabel('Number of Calls')
ax.set_title('Comparison of Calls Offered vs Calls Handled per Month')
ax.set_xticks(x)
ax.set_xticklabels(months, rotation=45)
ax.legend()
plt.show()
```



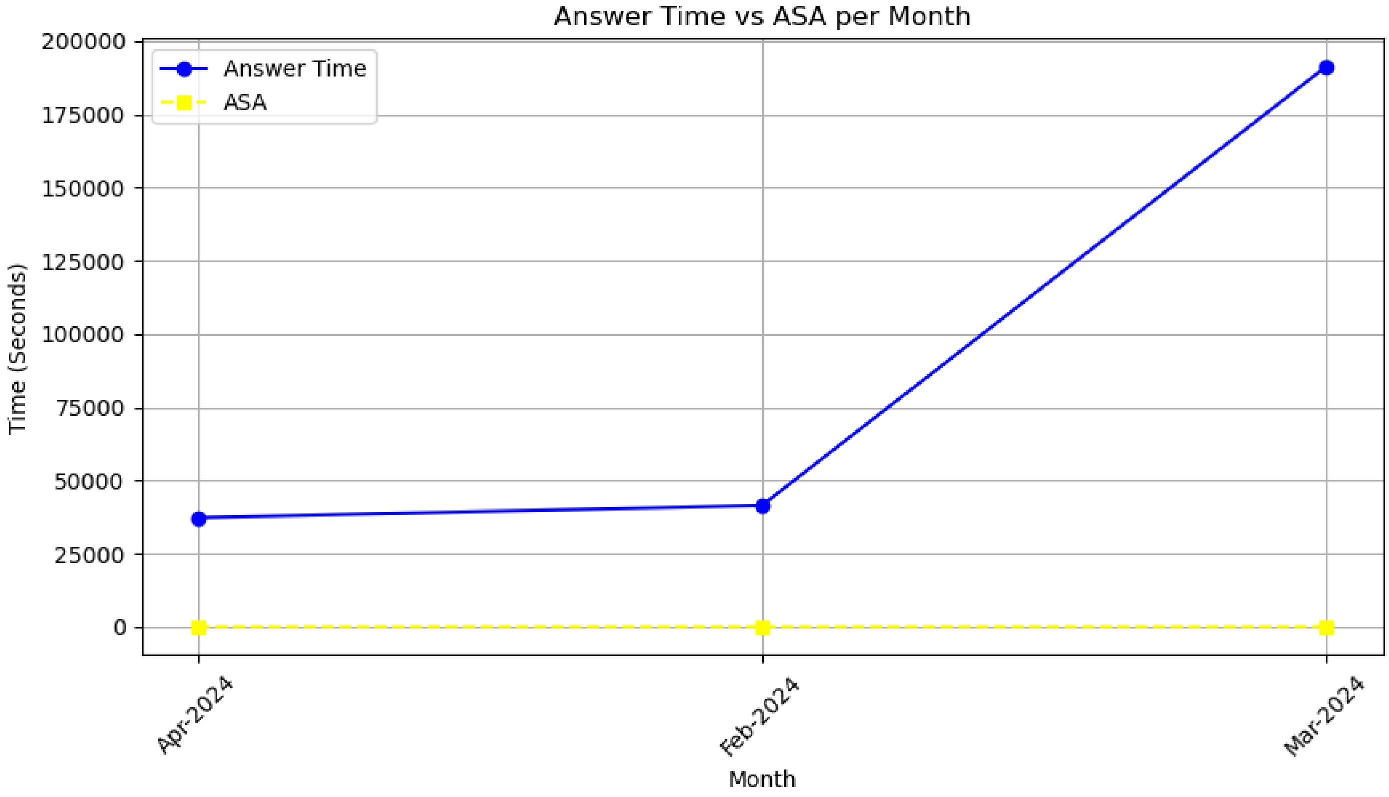
In [120]:

```
calls_data = df.groupby('Month')[['Calls Handled', 'Calls Abandon']].sum()

calls_data.plot(kind='bar', stacked=True, figsize=(10,5), color=['green', 'red'])
plt.title('Handled vs Abandoned Calls per Month')
plt.xlabel('Month')
plt.ylabel('Number of Calls')
plt.xticks(rotation=45)
plt.legend(['Calls Handled', 'Calls Abandon'])
plt.show()
```



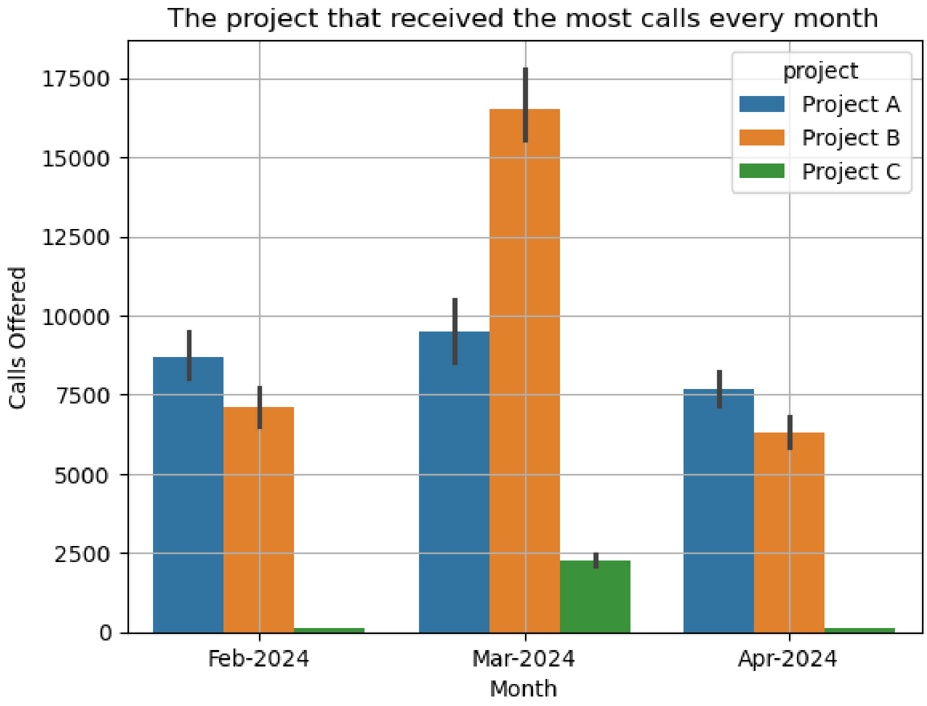
In [124]:

```
calls_data = df.groupby('Month')[['Answer Time', 'ASA']].mean()

plt.figure(figsize=(10,5))
plt.plot(calls_data.index, calls_data['Answer Time'], marker='o', linestyle='-', color='blue', label='Answer Time')
plt.plot(calls_data.index, calls_data['ASA'], marker='s', linestyle='--', color='yellow', label='ASA')

plt.xlabel('Month')
plt.ylabel('Time (Seconds)')
plt.title('Answer Time vs ASA per Month')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```



In [59]:

```
sns.barplot(data=df, x="Month", y="Calls Offered", hue="Project")
plt.title("The project that received the most calls every month")
plt.legend(title="project")
plt.grid()
```



In [49]:

```
monthly_data = df.groupby('Month').agg({
    'Calls Offered': 'sum',
    'Calls Handled': 'sum'
}).reset_index()

monthly_data['Handled Percentage'] = (monthly_data['Calls Handled'] / monthly_data['Calls Offered']) * 100

print(monthly_data)
```

```
      Month  Calls Offered  Calls Handled  Handled Percentage
0  Apr-2022         422469         420350           99.498425
1  Feb-2022         446217         444095           99.524447
2  Mar-2022         876199         857397           97.854140
```

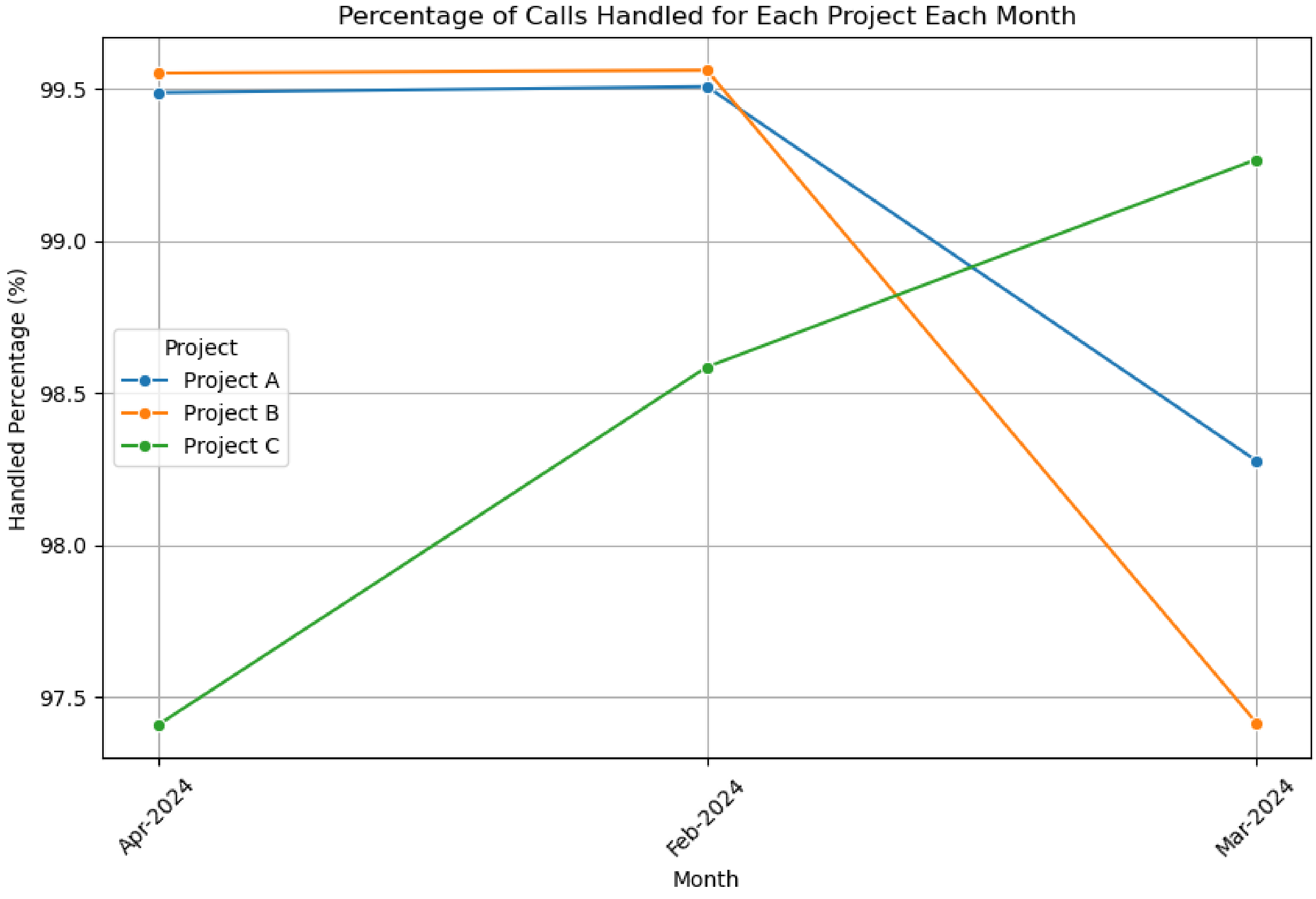
In [51]:

```
Projectly_data = df.groupby('Project').agg({
    'Calls Offered': 'sum',
    'Calls Handled': 'sum'
}).reset_index()

Projectly_data['Handled Percentage'] = (Projectly_data['Calls Handled'] / Projectly_data['Calls Offered']) * 100

print(Projectly_data)
```

```
    Project  Calls Offered  Calls Handled  Handled Percentage
0  Project A         767109         759676           99.031037
1  Project B         899756         884814           98.339328
2  Project C          78020          77352           99.143809
```

In [57]:

```
ProjectlyP_data = df.groupby(['Project', 'Month']).agg({
    'Calls Offered': 'sum',
    'Calls Handled': 'sum'
}).reset_index()


ProjectlyP_data['Handled Percentage'] = (ProjectlyP_data['Calls Handled'] / ProjectlyP_data['Calls Offered']) * 100

plt.figure(figsize=(10, 6))
sns.lineplot(data=ProjectlyP_data, x="Month", y="Handled Percentage", hue="Project", marker="o")

plt.title("Percentage of Calls Handled for Each Project Each Month")
plt.xlabel("Month")
plt.ylabel("Handled Percentage (%)")
plt.legend(title="Project")
plt.grid(True)
plt.xticks(rotation=45)

plt.show()
```

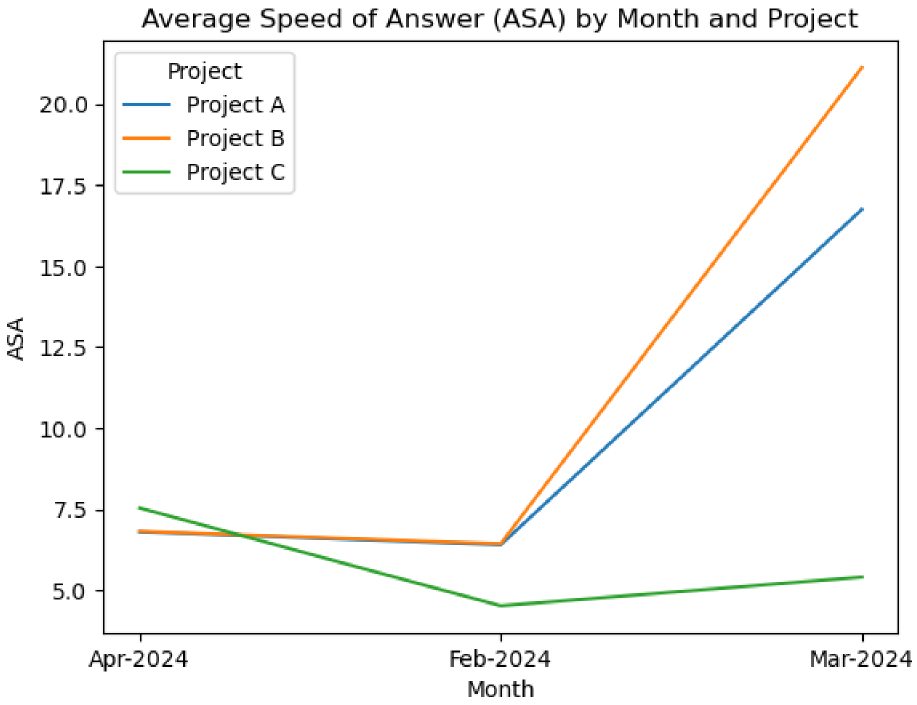


In [61]:

```
AsA = df.groupby(["Project", "Month"])["ASA"].mean().reset_index()
sns.lineplot(data=AsA, x="Month", y="ASA", hue="Project")
plt.title("Average Speed of Answer (ASA) by Month and Project")
```

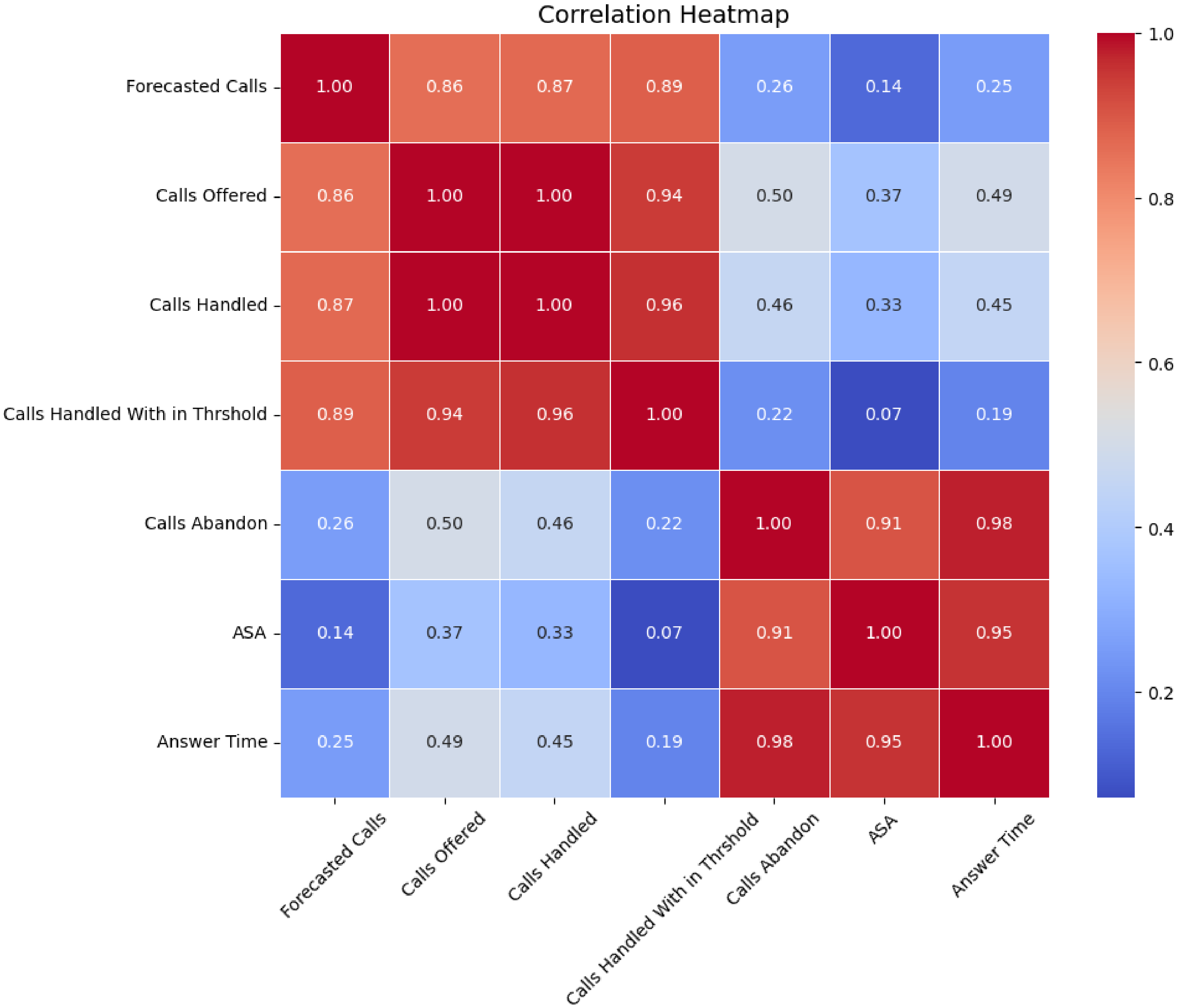
Out[61]:

```
Text(0.5, 1.0, 'Average Speed of Answer (ASA) by Month and Project')
```



In [67]:

```
numeric_df = df.select_dtypes(include=['number'])

corr = numeric_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr,
            xticklabels=corr.columns,
            yticklabels=corr.columns,
            cmap="coolwarm",
            annot=True,
            fmt=".2f",
            linewidths=0.5,
            cbar=True)

plt.title("Correlation Heatmap", fontsize=14)
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js