In [1]:

```python
import pandas as pd
from datetime import datetime

# Load your CSV data into a pandas DataFrame
file_path = r"C:\Users\mrmua\OneDrive\Desktop\Interview\Assignment_MySQ
df= pd.read_csv(file_path)

# Function to convert 12-hour AM/PM format to 24-hour format
def convert_to_24hr(time_str):
    return datetime.strptime(time_str, '%d-%b-%y %I:%M %p').strftime('%

# Apply conversion to 'Start' and 'End' columns
df['Start'] = df['Start'].apply(convert_to_24hr)
df['End'] = df['End'].apply(convert_to_24hr)

# Display the updated DataFrame
df
```

Out[1]:

	Name	Start	End	Activity
0	Priyanka	2023-10-10 22:43:00	2023-07-20 03:31:00	Inspection
1	Jyoti	2023-08-24 05:55:00	2023-05-17 20:19:00	Remote Inspection
2	Jyoti	2023-06-08 08:19:00	2023-04-08 05:55:00	Updates
3	Priyanka	2023-09-21 15:31:00	2023-05-27 10:43:00	Reporting
4	Priyanka	2023-10-07 03:31:00	2023-04-30 13:07:00	Reply to Customers
...	...	...	...	...
10880	Sharan	2023-04-27 20:19:00	2023-07-21 22:43:00	Reply to Customers
10881	Priyanka	2023-08-23 13:07:00	2023-08-11 10:43:00	Business Development
10882	Priyanka	2023-04-15 15:31:00	2023-06-29 15:31:00	Reporting
10883	Jyoti	2023-04-24 03:31:00	2023-06-16 20:19:00	Send Email
10884	Priyanka	2023-06-29 01:07:00	2023-08-13 08:19:00	Reply to Customers

10885 rows × 4 columns

In [2]:

```python
# Sort the dataframe by Name and Start time
df.sort_values(by=['Name', 'Start'], inplace=True)

# Initialize the result list
result = []
```

In [3]:

```
df
```

Out[3]:

	Name	Start	End	Activity
4341	Deepti	2023-03-29 15:31:00	2023-04-02 08:19:00	Business Development
10807	Deepti	2023-03-29 15:31:00	2023-04-20 15:31:00	Reply to Customers
9894	Deepti	2023-03-29 17:55:00	2023-05-13 08:19:00	Send Email
10846	Deepti	2023-03-29 17:55:00	2023-03-30 22:43:00	Fund raising
181	Deepti	2023-03-29 20:19:00	2023-08-11 10:43:00	Send Email
...	...	...	...	...
8831	Sharan	2023-10-15 03:31:00	2023-06-14 17:55:00	Reply to Customers
3600	Sharan	2023-10-15 08:19:00	2023-07-20 08:19:00	Fund raising
3683	Sharan	2023-10-15 08:19:00	2023-08-14 17:55:00	Reporting
6556	Sharan	2023-10-15 08:19:00	2023-04-08 05:55:00	Podcast
71	Sharan	2023-10-15 15:31:00	2023-07-23 17:55:00	Fund raising

10885 rows × 4 columns

In [4]:

```
# Process each bot's data
for name, group in df.groupby('Name'):
    group = group.sort_values(by='Start')
    current_start = None
    current_end = None
    activities = []

    for _, row in group.iterrows():
        if current_start is None:
            current_start = row['Start']
            current_end = row['End']
            activities = [row['Activity']]
        else:
            if row['Start'] <= current_end:  # Overlapping or contiguou
                current_end = max(current_end, row['End'])
                activities.append(row['Activity'])
            else:  # Non-overlapping period, save the current period an
                result.append([name, current_start, current_end, activi
                current_start = row['Start']
                current_end = row['End']
                activities = [row['Activity']]

    # Append the last period
    if current_start is not None:
        result.append([name, current_start, current_end, activities])
```

In [5]:

```
# Convert result to DataFrame
result_df = pd.DataFrame(result, columns=['Name', 'Start', 'End', 'Acti
```

In [6]:

```
result_df
```

Out[6]:

	Name	Start	End	Activities
0	Deepti	2023-03-29 15:31:00	2023-10-15 15:31:00	[Business Development, Reply to Customers, Sen...
1	Jyoti	2023-03-29 15:31:00	2023-10-15 13:07:00	[Business Development, Inspection, Fund raisin...
2	Priyanka	2023-03-29 15:31:00	2023-10-15 15:31:00	[Business Development, Remote Inspection, Podc...
3	Ravi	2023-03-29 15:31:00	2023-10-15 08:19:00	[Call, Call, Fund raising, Call, Fund raising,...
4	Ravi	2023-10-15 10:43:00	2023-06-28 13:07:00	[Reporting]
5	Ravi	2023-10-15 10:43:00	2023-07-26 01:07:00	[Inspection]
6	Ravi	2023-10-15 10:43:00	2023-06-10 17:55:00	[Send Email]
7	Ravi	2023-10-15 10:43:00	2023-07-06 13:07:00	[Inspection]
8	Ravi	2023-10-15 13:07:00	2023-04-18 10:43:00	[Send Email]
9	Ravi	2023-10-15 13:07:00	2023-08-27 08:19:00	[Podcast]
10	Sharan	2023-03-29 15:31:00	2023-10-15 05:55:00	[Inspection, Send Email, Fund raising, Updates...
11	Sharan	2023-10-15 08:19:00	2023-08-14 17:55:00	[Reporting]
12	Sharan	2023-10-15 08:19:00	2023-04-08 05:55:00	[Podcast]
13	Sharan	2023-10-15 08:19:00	2023-07-20 08:19:00	[Fund raising]
14	Sharan	2023-10-15 15:31:00	2023-07-23 17:55:00	[Fund raising]

In [7]:

```
# Output the result
output_path = r"C:\Users\mrmua\OneDrive\Desktop\Interview\Assignment_My
result_df.to_excel(output_path, index=False)
```

In [8]:

```
print(f"Aggregated data saved to {output_path}")
```

```
Aggregated data saved to C:\Users\mrmua\OneDrive\Desktop\Interview\Assignm
ent_MySQL_ecommerce\Aggregated_Time_Series.xlsx
```