Data Analyst Assignment

Name:- Abir Ghosh

Email:-abirghosh033@gmail.com

Contact No:- 9748248459

Github:- https://github.com/AGTech27/Browsing-Behavior-Analysis-Report.git

Browsing Behavior Analysis Report

1. Introduction

Project Overview

This report dives deep into user browsing behavior, analyzing web history logs to uncover patterns, engagement trends, and meaningful insights using Power BI and Python. The goal? To understand how users navigate the web and leverage that knowledge for better digital experiences.

Business Context

User behavior online is a goldmine of insights. By analyzing browsing history, we can identify peak activity times, user preferences, and common navigation paths. This helps businesses enhance user experiences, optimize content, and refine marketing strategies with data-driven decisions.

---

2. Data Cleaning & Preprocessing

Understanding the Dataset

The dataset consists of web history logs with key attributes like:

- **DeviceID:** Unique user session identifier.
- **URL:** The website accessed.
- **Eventtimeutc:** Timestamp of the visit (UTC format).
- **Transition**: Navigation type (e.g., link click, reload, direct entry, etc.).
- **Title:** Page title of the visited site.
- **VisitID:** Unique session identifier.
- **ReferringVisitID:** Previous session that led to the visit.

Data Preprocessing in Power BI

Step 1: Cleaning the Data

- Removed redundant metadata and extra headers.
- Dropped unnecessary columns to keep only relevant data.

Step 2: Convert Data Types

- Converted eventtimeutc into a structured **Date/Time** format.
- Changed visitID and referringVisitID into **Whole Numbers** for easier analysis.

Step 3: Extract New Features

- Extracted **Visit Date** and **Visit Hour** from eventtimeutc for time-based insights.

---

3. Exploratory Data Analysis & Visualizations

In [11]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from urllib.parse import urlparse

# Load the dataset, skipping metadata rows
file_path = "py_demo_client_extension_30_20250221075805 (1).csv"
df = pd.read_csv(file_path, skiprows=5)

# Identify correct column names
expected_columns = ["OrgId", "ParticipantId", "DeviceId", "URL", "EventTimeUTC", "Transition", "Title", "VisitId", "ReferringVisitId", "EventTime"]
df.columns = expected_columns

# Drop unnecessary columns and missing values
df = df.dropna(subset=["URL", "EventTimeUTC"])

# Convert EventTimeUTC to datetime
df["EventTimeUTC"] = pd.to_datetime(df["EventTimeUTC"], errors="coerce", utc=True)

# Extract domain from URL
df["Domain"] = df["URL"].apply(lambda x: urlparse(str(x)).netloc)

# Extract hour and day of the week
df["Hour"] = df["EventTimeUTC"].dt.hour
df["DayOfWeek"] = df["EventTimeUTC"].dt.day_name()

# Convert timestamp for session duration analysis
df["Timestamp"] = df["EventTimeUTC"].astype("int64") // 10**9

# Ensure consistency with existing analysis
# Assign cleaned data to df_corrected so that the rest of the code remains unchanged
df_corrected = df.copy()

# Display cleaned data info
print(df_corrected.info())
print(df_corrected.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5104 entries, 0 to 5103
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   OrgId             5104 non-null   object
 1   ParticipantId     5104 non-null   object
 2   DeviceId          5104 non-null   object
 3   URL               5104 non-null   object
 4   EventTimeUTC      5104 non-null   datetime64[ns, UTC]
 5   Transition        5104 non-null   object
 6   Title             5051 non-null   object
 7   VisitId           5104 non-null   int64
 8   ReferringVisitId  5104 non-null   int64
 9   EventTime         5104 non-null   object
 10  Domain            5104 non-null   object
 11  Hour              5104 non-null   int32
 12  DayOfWeek         5104 non-null   object
 13  Timestamp         5104 non-null   int64
dtypes: datetime64[ns, UTC](1), int32(1), int64(3), object(9)
memory usage: 538.4+ KB
None
```

	OrgId	ParticipantId	DeviceId	\
0	py_demo_client	demo	2nwjevbvxzm7ehb254	
1	py_demo_client	demo	2nwjevbvxzm7ehb254	
2	py_demo_client	demo	2nwjevbvxzm7ehb254	
3	py_demo_client	demo	2nwjevbvxzm7ehb254	
4	py_demo_client	demo	2nwjevbvxzm7ehb254	

	URL	\
0	chrome-extension://hkmmnfimlpcphpgnmgdecpdpaef...	
1	https://chromewebstore.google.com/detail/snaps...	
2	https://py-insights.com/account/demo/product?s...	
3	https://py-insights.com/account/demo/product?s...	
4	https://py-insights.com/account/demo/product	

	EventTimeUTC	Transition	Title	\
0	2025-02-21 07:58:02.688000+00:00	link	NaN	
1	2025-02-21 07:57:51.308000+00:00	link	Snapshot - Chrome Web Store	
2	2025-02-21 07:57:40.972000+00:00	link	PY Insights | Product	
3	2025-02-21 07:57:40.988000+00:00	link	PY Insights | Product	
4	2025-02-21 07:57:38.017000+00:00	link	PY Insights | Product	

	VisitId	ReferringVisitId	EventTime	\
0	166328	0	2025-02-20T23:58:02-08:00	
1	166327	166326	2025-02-20T23:57:51-08:00	
2	166319	0	2025-02-20T23:57:40-08:00	
3	166321	0	2025-02-20T23:57:40-08:00	
4	166318	0	2025-02-20T23:57:38-08:00	

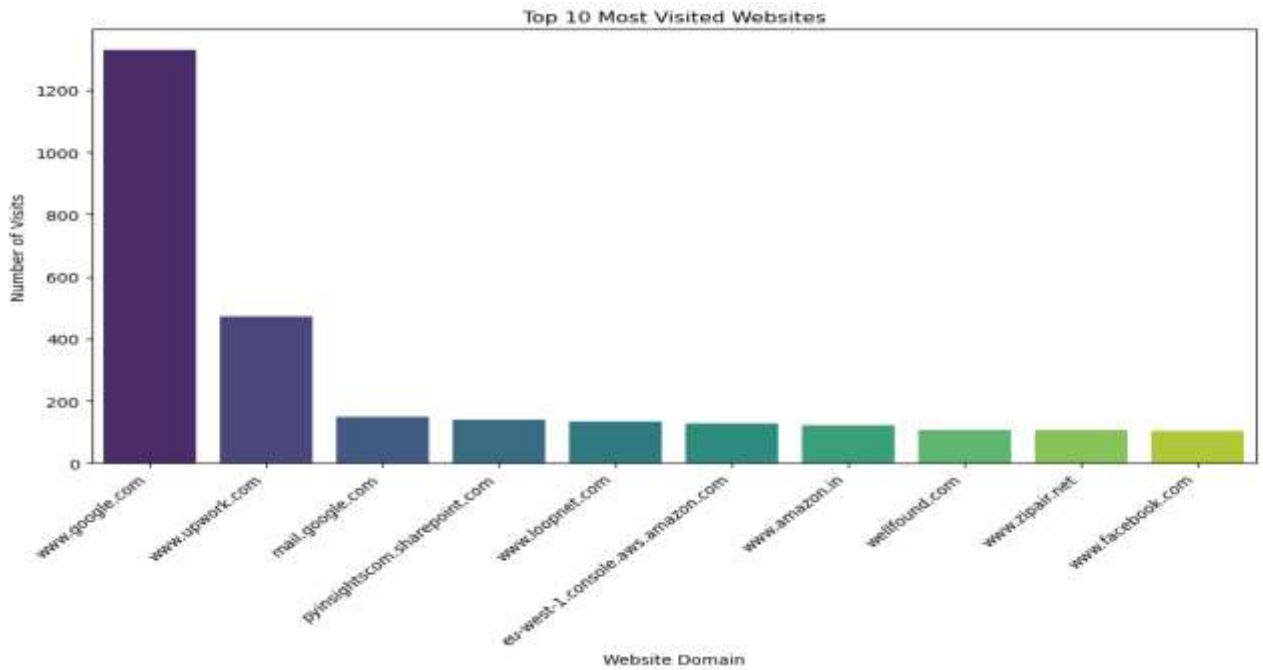
	Domain	Hour	DayOfWeek	Timestamp
0	hkmmnfimlpcphpgnmgdecpdpaefjnlga	7	Friday	1740124682
1	chromewebstore.google.com	7	Friday	1740124671
2	py-insights.com	7	Friday	1740124660
3	py-insights.com	7	Friday	1740124660
4	py-insights.com	7	Friday	1740124658

Browsing Trends by Hour

- **Goal:** Identify peak user activity hours to determine when users are most engaged online. Understanding this helps businesses schedule promotions, content releases, and marketing strategies effectively.
- **Visualization:** Clustered Column Chart, which visually represents usage spikes across different hours of the day.
- **Key Takeaway:** Businesses can use this data to optimize digital engagement, such as posting social media content or running targeted ads during peak hours.

In [42]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from urllib.parse import urlparse
```

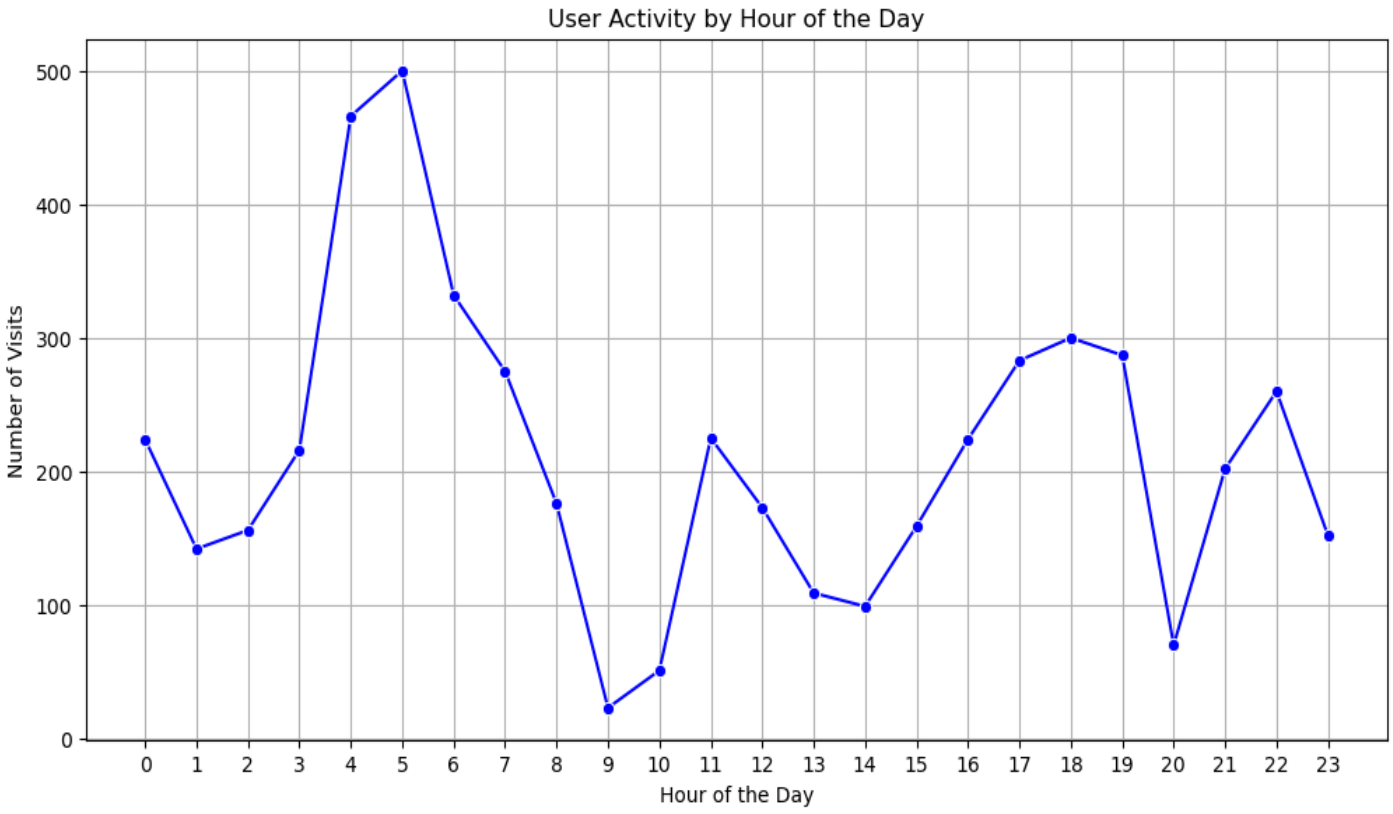
```
# Get the top 10 most visited websites
top_sites = df_corrected["Domain"].value_counts().head(10)

# Plot the top websites
plt.figure(figsize=(12, 6))
sns.barplot(x=top_sites.index, y=top_sites.values, palette="viridis")
plt.xticks(rotation=45, ha="right")
plt.xlabel("Website Domain")
plt.ylabel("Number of Visits")
plt.title("Top 10 Most Visited Websites")
plt.show()
```



Daily Browsing Patterns

- **Goal:** Analyze user activity variations across different days to detect browsing habits and weekly trends.
- **Visualization:** Line Chart, offering a clear trend analysis over time.
- **Key Takeaway:** This insight is valuable for campaign planning, ensuring that businesses reach their audience on the most active days.

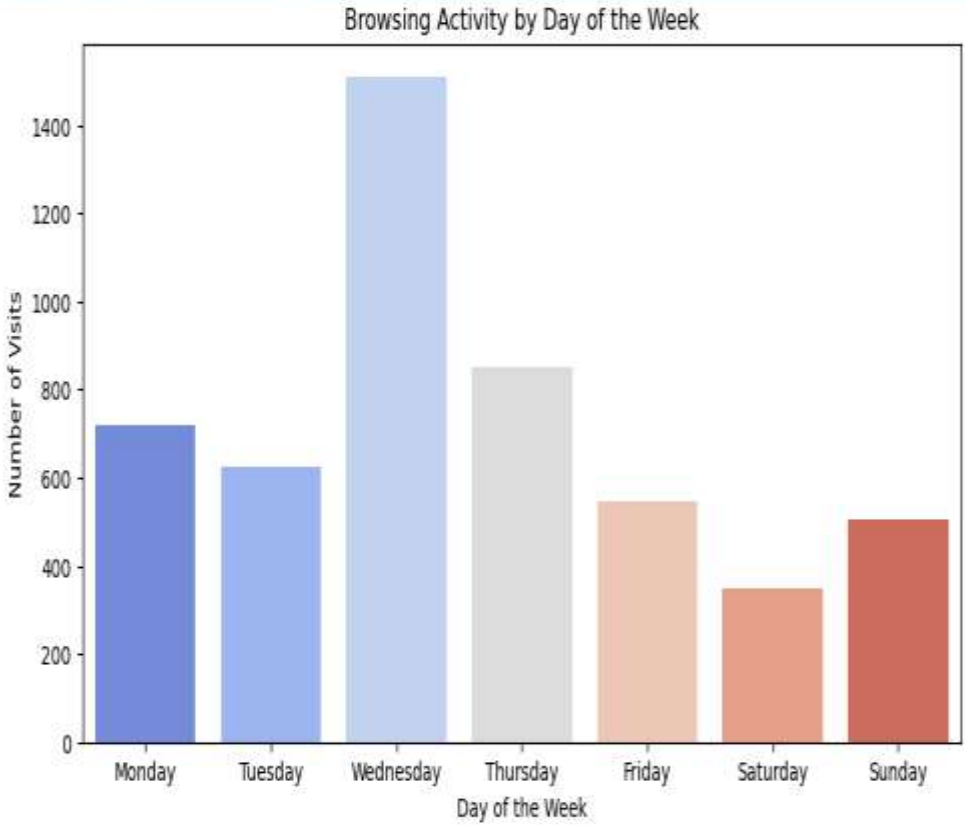
In [43]:
```
# Extract hour of visit from the timestamp
df_corrected["Hour"] = df_corrected["EventTimeUTC"].dt.hour

# Aggregate visits by hour
hourly_visits = df_corrected["Hour"].value_counts().sort_index()

# Plot hourly activity
plt.figure(figsize=(12, 6))
sns.lineplot(x=hourly_visits.index, y=hourly_visits.values, marker="o", color="blue")
plt.xticks(range(0, 24))
plt.xlabel("Hour of the Day")
plt.ylabel("Number of Visits")
plt.title("User Activity by Hour of the Day")
plt.grid(True)
plt.show()
```



Most Visited Websites

- **Goal:** Identify which websites users frequent the most, helping businesses understand user interests.
- **Visualization:** Bar Chart displaying the Top 10 most visited websites.
- **Key Takeaway:** Companies can use this information for targeted advertising, content partnerships, and trend analysis in user preferences.

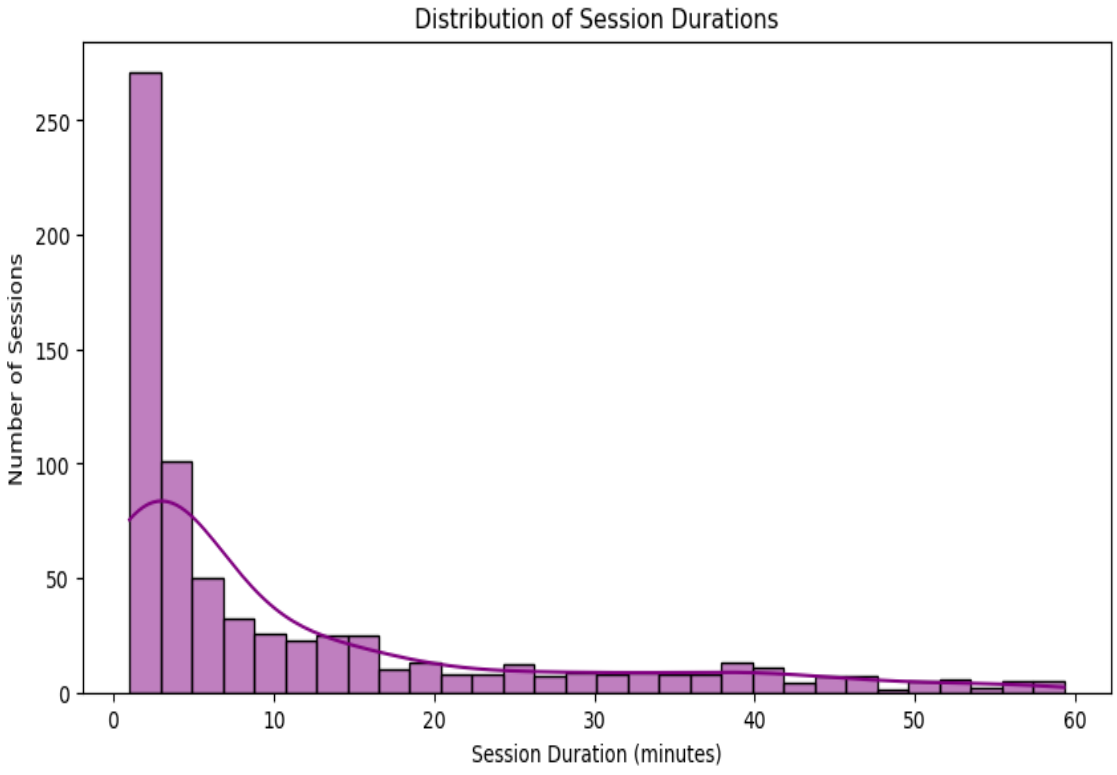
In [44]:
```
# Extract day of the week
df_corrected["DayOfWeek"] = df_corrected["EventTimeUTC"].dt.day_name()

# Aggregate visits by day of the week
daily_visits = df_corrected["DayOfWeek"].value_counts()[
    ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
]

# Plot daily activity
plt.figure(figsize=(10, 5))
sns.barplot(x=daily_visits.index, y=daily_visits.values, palette="coolwarm")
plt.xlabel("Day of the Week")
plt.ylabel("Number of Visits")
plt.title("Browsing Activity by Day of the Week")
plt.show()
```



Interaction Type Distribution

- **Goal**: Understand how users navigate different websites—whether they click on internal links, reload pages, or manually type URLs.
- **Visualization:** Stacked Bar Chart, categorizing various navigation methods.
- **Key Takeaway:** This information helps optimize website UI/UX by focusing on elements that keep users engaged and reducing friction in navigation.

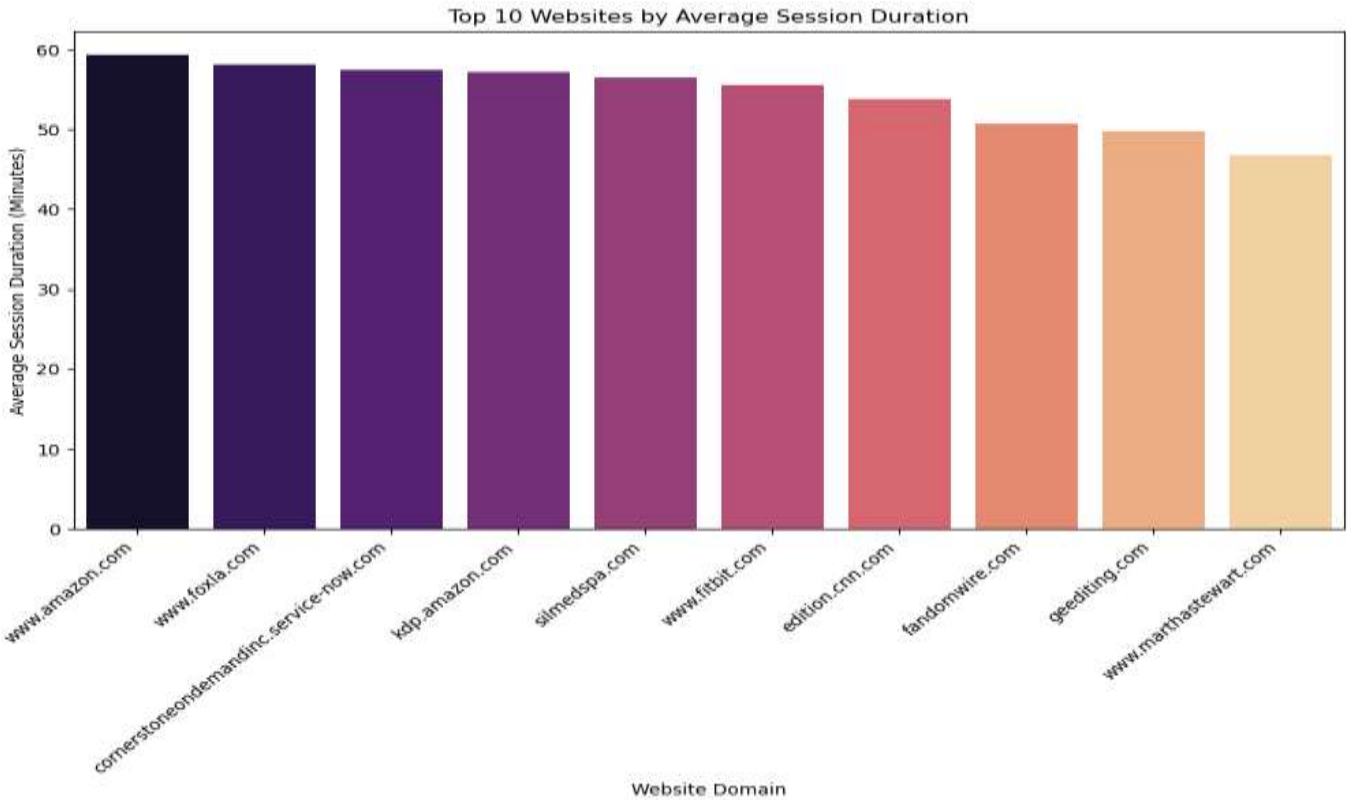
In [45]:
```
# Convert EventTimeUTC to seconds for session duration calculation
df_corrected["Timestamp"] = df_corrected["EventTimeUTC"].astype("int64") // 10**9

# Sort data by DeviceId and Timestamp
df_corrected = df_corrected.sort_values(by=["DeviceId", "Timestamp"])

# Compute session duration (time between consecutive visits for the same user)
df_corrected["SessionDuration"] = df_corrected.groupby("DeviceId")["Timestamp"].diff()

# Convert to minutes
df_corrected["SessionDuration"] = df_corrected["SessionDuration"] / 60

# Remove outliers (e.g., gaps longer than 60 minutes)
df_filtered = df_corrected[df_corrected["SessionDuration"].between(1, 60)]

# Plot session duration distribution
plt.figure(figsize=(10, 5))
sns.histplot(df_filtered["SessionDuration"], bins=30, kde=True, color="purple")
plt.xlabel("Session Duration (minutes)")
plt.ylabel("Number of Sessions")
plt.title("Distribution of Session Durations")
plt.show()
```



- ## Top 10 Websites by Average Session Duration

- **Goal:** Determine where users spend the most time to understand engagement levels.
- **Visualization:** Stacked Column Chart to compare session durations across top websites.
- **Key Takeaway:**
  - **Amazon** and major e-commerce platforms lead in user engagement due to extensive browsing time.
  - **News and service-based websites** retain users for longer durations as they consume content.
  - **Niche content platforms** attract users seeking specific, in-depth information.
  - **Optimizing UI/UX** for these high-retention websites can further enhance user experience and engagement.

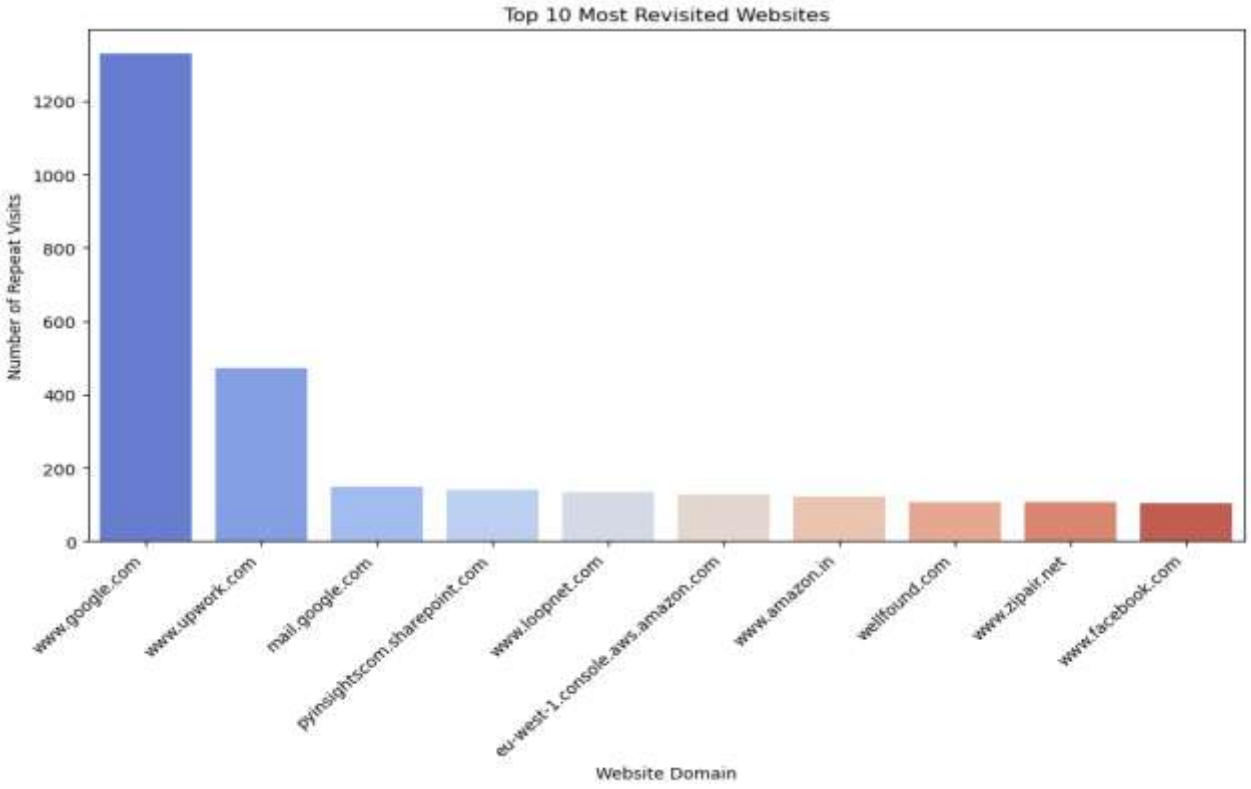
In [46]:
```
# Calculate average session duration per website
avg_session_duration = df_filtered.groupby("Domain")["SessionDuration"].mean().dropna().sort_values(ascending=False).head(10)

# Plot bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x=avg_session_duration.index, y=avg_session_duration.values, palette="magma")
plt.xticks(rotation=45, ha="right")
plt.xlabel("Website Domain")
plt.ylabel("Average Session Duration (Minutes)")
plt.title("Top 10 Websites by Average Session Duration")
plt.show()
```



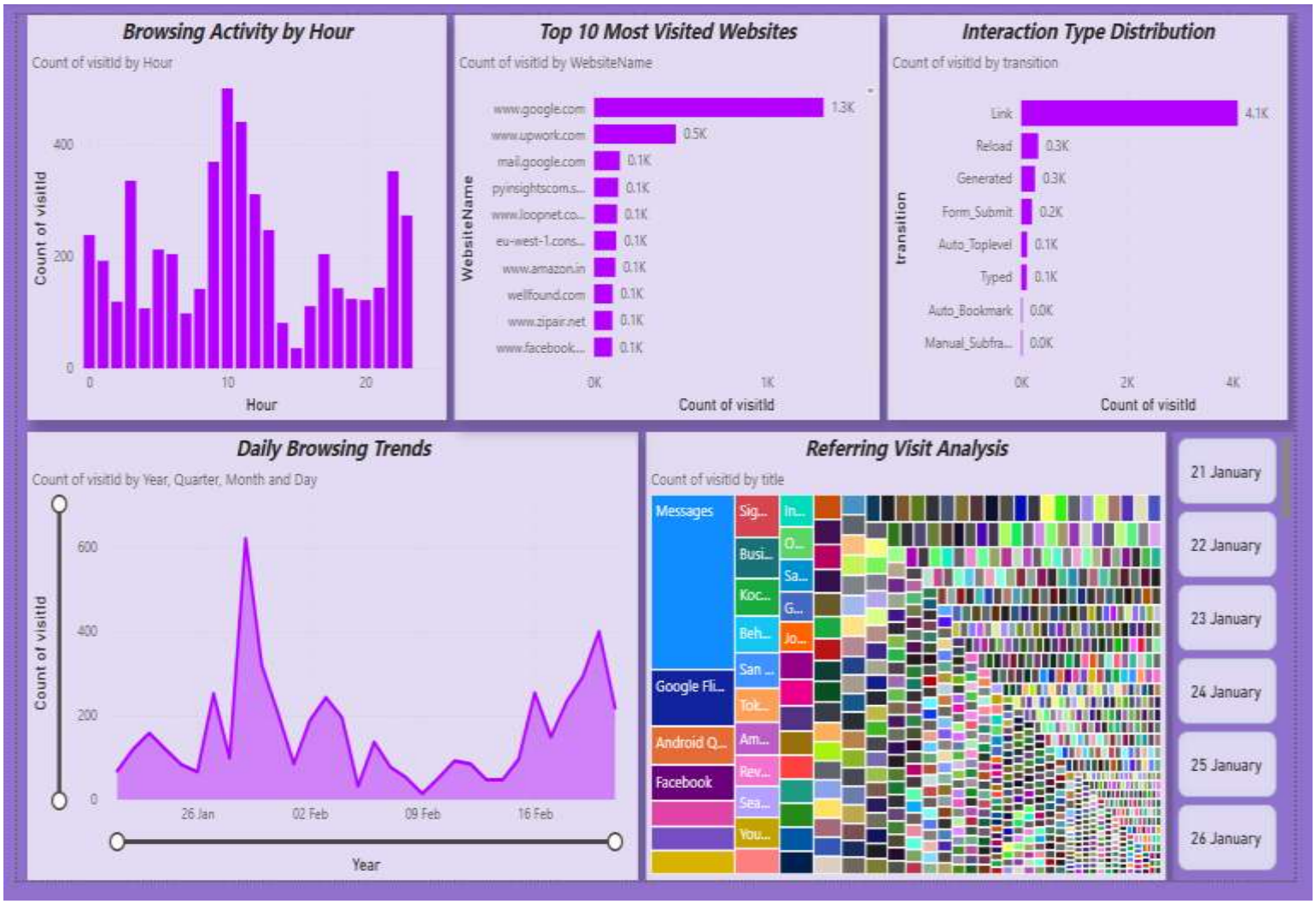
Top 10 Most Revisited Websites

- **Goal:** Track how users return to certain websites repeatedly and identify the most frequently revisited domains.
- **Visualization:** Tree Map, showcasing the most revisited websites based on repeat visit counts.
- **Key Takeaway:** Businesses can use this data to improve content linking strategies, retain returning visitors, and enhance website traffic flow by strategically placing recommended links.

In [47]:
```
# Count repeat visits per website
repeat_visits = df_corrected["Domain"].value_counts().head(10)

# Plot bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x=repeat_visits.index, y=repeat_visits.values, palette="coolwarm")
plt.xticks(rotation=45, ha="right")
plt.xlabel("Website Domain")
plt.ylabel("Number of Repeat Visits")
plt.title("Top 10 Most Revisited Websites")
plt.show()
```



Power BI Dashboard



Insights from Power BI Dashboard

- **Users Have a Preferred "Prime Time" for Browsing**
    - The data shows that users are most active during specific hours, likely in the evenings when they have free time. This suggests businesses should time their content releases, social media posts, and ads around these peak hours for maximum engagement.
- **People Revisit the Same Websites Regularly**
    - There's a strong pattern of users revisiting certain sites repeatedly. This could mean they trust these sources, find value in the content, or are making repeated purchase decisions. Companies can leverage this by offering loyalty programs or retargeting ads to keep users engaged.
- **Navigation Habits Show Users Prefer Clicks Over Manual Typing**
    - A large percentage of users access websites through internal links rather than typing the URL. This highlights the importance of **intuitive website design and easy navigation**—businesses should ensure key pages are easily accessible through clear CTAs and internal linking strategies.
- **Session Durations Reveal Engagement Levels**
    - Some websites have significantly higher session durations, meaning users are deeply engaged. These could be e-commerce sites where users browse multiple products, news portals where they read articles, or educational platforms. Businesses should focus on keeping users engaged with interactive content, recommendations, and personalized experiences.
- **Not All Visits Lead to Immediate Actions**
    - Just because a user visits a website doesn't mean they take action right away. Many return later, possibly after research or comparison shopping. Marketers should use retargeting ads and follow-up emails to capture these users at the right moment.
- **Opportunities for Better User Retention**
    - Websites that don't see repeat visits might need better content strategies. Improving engagement through blog articles, newsletters, or push notifications can help turn one-time visitors into loyal users.
- **Marketing Should Align With Browsing Trends**
    - Instead of blindly scheduling posts or campaigns, businesses can **align their marketing with user behavior trends**—posting at peak hours, tailoring content to frequently visited sites, and optimizing for platforms where users spend the most time.

Key Insights & Strategic Recommendations

Key Findings

- **Peak Browsing Hours:** Users are most active during specific time windows—ideal for engagement strategies.
- **Frequent Website Visits:** Users consistently visit certain sites, revealing strong preferences.
- **User Navigation Trends:** Most users rely on internal links rather than typing URLs manually.
- **Browsing Journeys:** Revisit data helps track user movement across the web.

Recommendations

- **Personalized Experiences:** Tailor content based on browsing patterns to increase engagement.
- **Enhanced Site Navigation:** Since most users rely on internal links, improving UI/UX can boost retention.
- **Targeted Marketing:** Use browsing trends to schedule ads and promotions for maximum impact.
- **Content Optimization:** Websites with higher session durations can focus on maintaining engaging content, while lower retention sites should enhance their user experience.

Conclusion

This analysis sheds light on user browsing behavior, offering insights that help businesses enhance digital strategies. Power BI's interactive visualizations reveal browsing trends, peak activity times, and user interaction patterns—key elements for informed decision-making.

Key Takeaways:

- Users have specific prime times for browsing, making it essential to schedule content strategically.
- People revisit certain websites regularly, which means businesses should focus on retaining loyal users.
- Navigation trends show that users prefer internal links over manually typing URLs, emphasizing the need for a smooth user experience.
- Not all visits lead to immediate actions—remarketing and follow-ups can help capture returning users.