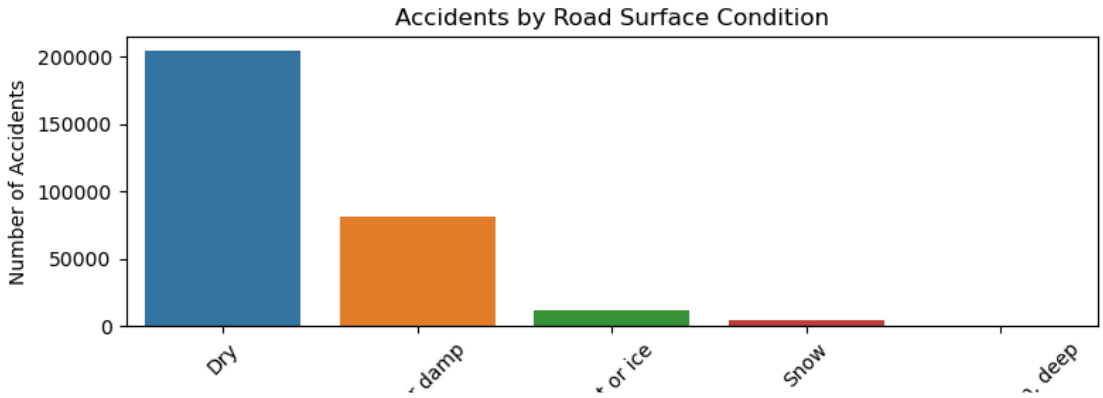
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

# File path
file_path = r"C:/Users/nidhi/Downloads/Road Accident Data (1).csv"

# Check if file exists
if not os.path.exists(file_path):
    print(f"❌ File not found: {file_path}")
else:
    # Load dataset
    df = pd.read_csv(file_path)

    # Show available columns
    print("✅ Columns in dataset:\n", df.columns.tolist())

    # Drop missing values from important columns
    df.dropna(subset=['Road_Surface_Conditions', 'Weather_Conditions', 'Time

    # Accidents by Road Condition
    road_condition_counts = df['Road_Surface_Conditions'].value_counts()
    plt.figure(figsize=(8, 4))
    sns.barplot(x=road_condition_counts.index, y=road_condition_counts.value
    plt.title("Accidents by Road Surface Condition")
    plt.xlabel("Road Surface Condition")
    plt.ylabel("Number of Accidents")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()


    # Accidents by Weather Condition
    weather_counts = df['Weather_Conditions'].value_counts()
    plt.figure(figsize=(8, 4))
    sns.barplot(x=weather_counts.index, y=weather_counts.values)
    plt.title("Accidents by Weather Condition")
    plt.xlabel("Weather Condition")
    plt.ylabel("Number of Accidents")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()


    # Accidents by Time of Day
    df['Hour'] = pd.to_datetime(df['Time'], format='%H:%M', errors='coerce')
    df = df.dropna(subset=['Hour'])
    df['Hour'] = df['Hour'].astype(int)

    hourly_accidents = df['Hour'].value_counts().sort_index()
    plt.figure(figsize=(10, 4))
    sns.lineplot(x=hourly_accidents.index, y=hourly_accidents.values, marker
    plt.title("Accidents by Hour of Day")
    plt.xlabel("Hour of Day")
    plt.ylabel("Number of Accidents")
    plt.xticks(range(0, 24))
    plt.tight_layout()
    plt.show()
```

```
# Accident Hotspots (Geospatial)
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['Longitude'], y=df['Latitude'], alpha=0.3)
plt.title('Accident Hotspots')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.tight_layout()
plt.show()


# Correlation Matrix (Contributing Factors)
factors = ['Road_Surface_Conditions', 'Weather_Conditions', 'Hour']
factor_data = pd.get_dummies(df[factors], drop_first=True)
corr_matrix = factor_data.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=False, cmap='coolwarm', cbar=True)
plt.title("Correlation Between Contributing Factors")
plt.tight_layout()
plt.show()
```

```
✅ Columns in dataset:
 ['Accident_Index', 'Accident Date', 'Month', 'Day_of_Week', 'Year', 'Ju
nction_Control', 'Junction_Detail', 'Accident_Severity', 'Latitude', 'Li
ght_Conditions', 'Local_Authority_(District)', 'Carriageway_Hazards', 'L
ongitude', 'Number_of_Casualties', 'Number_of_Vehicles', 'Police_Force',
'Road_Surface_Conditions', 'Road_Type', 'Speed_limit', 'Time', 'Urban_or
_Rural_Area', 'Weather_Conditions', 'Vehicle_Type']
```



In [ ]: