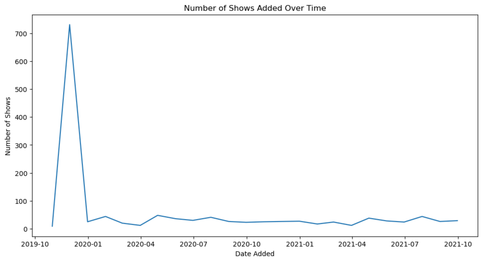
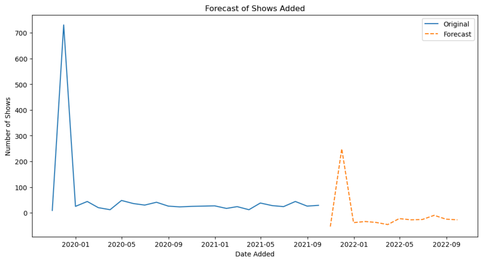
In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.metrics import mean_squared_error

# Load the dataset
disney_data = pd.read_csv('C:/Users/pujit/Downloads/disney_plus_titles.csv')

# Preprocess the date_added column to datetime format
disney_data['date_added'] = pd.to_datetime(disney_data['date_added'], errors='coerce')

# Drop rows with NaN values in date_added
disney_data = disney_data.dropna(subset=['date_added'])

# Set the date_added column as the index
disney_data.set_index('date_added', inplace=True)

# Plot the number of shows added over time
shows_over_time = disney_data['show_id'].resample('M').count()

plt.figure(figsize=(12, 6))
plt.plot(shows_over_time)
plt.title('Number of Shows Added Over Time')
plt.xlabel('Date Added')
plt.ylabel('Number of Shows')
plt.show()

# Exponential Smoothing Model
model = ExponentialSmoothing(shows_over_time, trend='add', seasonal='add', seasonal_periods=12)
model_fit = model.fit()
forecast = model_fit.forecast(steps=12)

# Plot the forecast
plt.figure(figsize=(12, 6))
plt.plot(shows_over_time, label='Original')
plt.plot(forecast, label='Forecast', linestyle='--')
plt.title('Forecast of Shows Added')
plt.xlabel('Date Added')
plt.ylabel('Number of Shows')
plt.legend()
plt.show()

# Evaluate the model
mse = mean_squared_error(shows_over_time[-12:], forecast)
print(f'Mean Squared Error: {mse}')
```



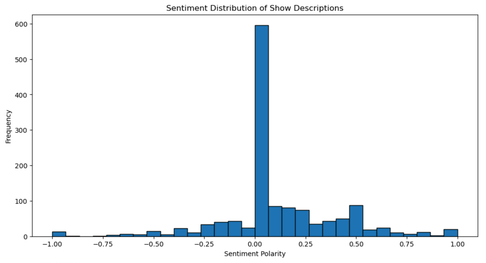



```
Mean Squared Error: 7300.912349786591
```

In [7]:
```
pip install textblob
```

```
Collecting textblob
  Obtaining dependency information for textblob from https://files.pythonhosted.org/packages/02/07/5fd2945356dd839974d3a25de8a142dc37293c21315729a41e775b5f3569/textblob-0.18.0.post0-py3-none-any.whl.metadata
  Downloading textblob-0.18.0.post0-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: nltk>=3.8 in c:\users\pujit\anaconda3\lib\site-packages (from textblob) (3.8.1)
Requirement already satisfied: click in c:\users\pujit\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (8.0.4)
Requirement already satisfied: joblib in c:\users\pujit\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\pujit\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\pujit\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (4.65.0)
Requirement already satisfied: colorama in c:\users\pujit\anaconda3\lib\site-packages (from click->nltk>=3.8->textblob) (0.4.6)
Downloading textblob-0.18.0.post0-py3-none-any.whl (626 kB)
   ---------------------------------------- 0.0/626.3 kB ? eta -:--:--
    --------------------------------------- 10.2/626.3 kB ? eta -:--:--
   - -------------------------------------- 30.7/626.3 kB 330.3 kB/s eta 0:00:02
   - -------------------------------------- 30.7/626.3 kB 330.3 kB/s eta 0:00:02
   -- ------------------------------------- 61.4/626.3 kB 328.2 kB/s eta 0:00:02
   ----- ---------------------------------- 92.2/626.3 kB 438.1 kB/s eta 0:00:02
   ----- ---------------------------------- 92.2/626.3 kB 438.1 kB/s eta 0:00:02
   ------- -------------------------------- 122.9/626.3 kB 400.9 kB/s eta 0:00:02
   ------- -------------------------------- 174.1/626.3 kB 499.5 kB/s eta 0:00:01
   --------- ------------------------------ 174.1/626.3 kB 499.5 kB/s eta 0:00:01
   ---------- ----------------------------- 225.3/626.3 kB 529.7 kB/s eta 0:00:01
   ---------- ----------------------------- 286.7/626.3 kB 589.5 kB/s eta 0:00:01
   ------------ --------------------------- 317.4/626.3 kB 595.3 kB/s eta 0:00:01
   ------------ --------------------------- 419.8/626.3 kB 708.6 kB/s eta 0:00:01
   ------------- -------------------------- 471.0/626.3 kB 775.8 kB/s eta 0:00:01
   ------------- -------------------------- 471.0/626.3 kB 775.8 kB/s eta 0:00:01
   ---------------- ----------------------- 512.0/626.3 kB 713.7 kB/s eta 0:00:01
   ---------------- ----------------------- 563.2/626.3 kB 721.9 kB/s eta 0:00:01
   ----------------- ---------------------- 624.6/626.3 kB 756.6 kB/s eta 0:00:01
   ---------------------------------------- 626.3/626.3 kB 744.1 kB/s eta 0:00:00
Installing collected packages: textblob
Successfully installed textblob-0.18.0.post0
Note: you may need to restart the kernel to use updated packages.
```

In [8]:
```
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt

# Load the dataset
disney_data = pd.read_csv('C:/Users/pujit/Downloads/disney_plus_titles.csv')

# Function to compute sentiment polarity
def compute_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity

# Apply sentiment analysis
disney_data['sentiment'] = disney_data['description'].apply(compute_sentiment)

# Plot the sentiment distribution
plt.figure(figsize=(12, 6))
plt.hist(disney_data['sentiment'], bins=30, edgecolor='k')
plt.title('Sentiment Distribution of Show Descriptions')
plt.xlabel('Sentiment Polarity')
plt.ylabel('Frequency')
plt.show()

# Show some statistics about the sentiment
sentiment_stats = disney_data['sentiment'].describe()
print(sentiment_stats)
```




```
count    1368.000000
mean        0.098366
std         0.296196
min        -1.000000
25%         0.000000
50%         0.000000
75%         0.250000
max         1.000000
Name: sentiment, dtype: float64
```

In [9]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Load the dataset
disney_data = pd.read_csv("C:/Users/pujit/Downloads/disney_plus_titles.csv")

# Select relevant features
features = disney_data[['type', 'rating', 'duration', 'listed_in']]

# Preprocess the duration feature
def preprocess_duration(duration):
    if 'Season' in duration:
        return int(duration.split(' ')[0]) * 365  # Approximate a season to a year
    else:
        return int(duration.split(' ')[0])

features['duration'] = features['duration'].apply(preprocess_duration)

# Handle missing values by filling with the mode
features = features.fillna(features.mode().iloc[0])

# One-hot encode categorical variables
encoder = OneHotEncoder(sparse=False)
encoded_features = encoder.fit_transform(features[['type', 'rating', 'listed_in']])
encoded_features_df = pd.DataFrame(encoded_features, columns=encoder.get_feature_names_out(['type', 'rating', 'listed_in']))

# Combine encoded features with the numerical feature
processed_features = pd.concat([encoded_features_df, features[['duration']].reset_index(drop=True)], axis=1)

# Standardize the features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(processed_features)

# Apply K-means clustering
kmeans = KMeans(n_clusters=5, random_state=42)
clusters = kmeans.fit_predict(scaled_features)

# Add cluster labels to the original dataset
disney_data['cluster'] = clusters

# Use PCA for 2D visualization of the clusters
pca = PCA(n_components=2)
pca_features = pca.fit_transform(scaled_features)

# Plot the clusters
plt.figure(figsize=(12, 6))
plt.scatter(pca_features[:, 0], pca_features[:, 1], c=clusters, cmap='viridis', marker='o', edgecolor='k')
plt.title('Clusters of Shows')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(label='Cluster')
plt.show()
```

```
C:\Users\pujit\AppData\Local\Temp\ipykernel_17724\4807174482.py:28: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  features['duration'] = features['duration'].apply(preprocess_duration)
C:\Users\pujit\anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be removed in 1.4.  `sparse_output` is ignored unless you leave `sparse` to its default value.
  warnings.warn(
C:\Users\pujit\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```


In [ ]:
```
```