*** *SECTION-1* ***

*** *NETFLIX DATA SET* ***

*** *Exploratory Data Analysis (EDA)* ***

Step 1: Import necessary libraries

- Pandas
- Matplotlib
- Seaborn

In [4]:
```
  # Pandas is a powerful, open-source library in Python for data manipulation an

import pandas as pd

  # Matplotlib is a popular, open-source plotting library for Python, providing

import matplotlib.pyplot as plt

  # Seaborn is a Python data visualization library built on top of Matplotlib, p

import seaborn as sns
```

Step 2: Load Netflix dataset

In [6]:
```
netflix_data = pd.read_csv('netflix_data.csv')   # Replace 'netflix_data.csv' wi
netflix_data
```

Out[6]:

	Title	Genre	Tags	Languages	Series or Movie	Hidd Ge Sco
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Comedy Programmes,Romantic TV Comedies,Horror ...	Swedish, Spanish	Series	4
1	HOW TO BUILD A GIRL	Comedy	Dramas,Comedies,Films Based on Books,British	English	Movie	7
2	The Con-Heartist	Comedy, Romance	Romantic Comedies,Comedies,Romantic Films,Thai...	Thai	Movie	8
3	Gleboka woda	Drama	TV Dramas,Polish TV Shows,Social Issue TV Dramas	Polish	Series	8
4	Only a Mother	Drama	Social Issue Dramas,Dramas,Movies Based on Boo...	Swedish	Movie	8
...	...	...	...	...	...	
9420	13 Going on 30	Comedy, Fantasy, Romance	Romantic Comedies,Comedies,Romantic Films,Roma...	English, Portuguese	Movie	3
9421	LIFE 2.0	Documentary	Social & Cultural Documentaries,Biographical D...	English	Movie	8
9422	Brand New Day	Documentary, Music	Australian Comedies,Romantic Comedies,Australi...	English	Movie	8
9423	Daniel Arends: Blessuretijd	Comedy	Stand-up Comedy,International Movies,Comedies	Dutch	Movie	8

	Title	Genre	Tags	Languages	Series or Movie	Hidd Ge Sco
9424	DreamWorks Happy Holidays from Madagascar	Animation, Comedy, Family	TV Comedies,Kids TV,Animal Tales,TV Cartoons,T...	English	Series	8

9425 rows × 29 columns

Step 3: Get basic information about the dataset

1. head() Method

In [9]:
```
print("\nBasic Information about the Dataset:\n")
netflix_data.head()   # Displays the first few rows of the dataset.
```

```
Basic Information about the Dataset:
```

Out[9]:

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score	
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Comedy Programmes,Romantic TV Comedies,Horror ...	Swedish, Spanish	Series	4.3	
1	HOW TO BUILD A GIRL	Comedy	Dramas,Comedies,Films Based on Books,British	English	Movie	7.0	
2	The Con-Heartist	Comedy, Romance	Romantic Comedies,Comedies,Romantic Films,Thai...	Thai	Movie	8.6	
3	Gleboka woda	Drama	TV Dramas,Polish TV Shows,Social Issue TV Dramas	Polish	Series	8.7	
4	Only a Mother	Drama	Social Issue Dramas,Dramas,Movies Based on Boo...	Swedish	Movie	8.3	Lithuania

5 rows × 29 columns

2. info() Method

In [11]:
```
netflix_data.info()   # Provides information about dataset shape, column data ty
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9425 entries, 0 to 9424
Data columns (total 29 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Title                  9425 non-null   object 
 1   Genre                  9400 non-null   object 
 2   Tags                   9389 non-null   object 
 3   Languages              9255 non-null   object 
 4   Series or Movie        9425 non-null   object 
 5   Hidden Gem Score       9415 non-null   float64
 6   Country Availability   9414 non-null   object 
 7   Runtime                9424 non-null   object 
 8   Director               7120 non-null   object 
 9   Writer                 7615 non-null   object 
 10  Actors                 9314 non-null   object 
 11  View Rating            6827 non-null   object 
 12  IMDb Score             9417 non-null   float64
 13  Rotten Tomatoes Score  5445 non-null   float64
 14  Metacritic Score       4082 non-null   float64
 15  Awards Received        5226 non-null   float64
 16  Awards Nominated For   6376 non-null   float64
 17  Boxoffice              3754 non-null   object 
 18  Release Date           9217 non-null   object 
 19  Netflix Release Date   9425 non-null   object 
 20  Production House       4393 non-null   object 
 21  Netflix Link           9425 non-null   object 
 22  IMDb Link              9101 non-null   object 
 23  Summary                9420 non-null   object 
 24  IMDb Votes             9415 non-null   float64
 25  Image                  9425 non-null   object 
 26  Poster                 8487 non-null   object 
 27  TMDb Trailer           9425 non-null   object 
 28  Trailer Site           9424 non-null   object 
dtypes: float64(7), object(22)
memory usage: 2.1+ MB
```

3. shape Method

In [13]:
```
netflix_data.shape  # Returns the number of rows and columns.
```

Out[13]:
```
(9425, 29)
```

4. columns Method

In [15]:
```
netflix_data.columns  # Displays column names.
```

Out[15]:
```
Index(['Title', 'Genre', 'Tags', 'Languages', 'Series or Movie',
       'Hidden Gem Score', 'Country Availability', 'Runtime', 'Director',
       'Writer', 'Actors', 'View Rating', 'IMDb Score',
       'Rotten Tomatoes Score', 'Metacritic Score', 'Awards Received',
       'Awards Nominated For', 'Boxoffice', 'Release Date',
       'Netflix Release Date', 'Production House', 'Netflix Link', 'IMDb Link',
       'Summary', 'IMDb Votes', 'Image', 'Poster', 'TMDb Trailer',
       'Trailer Site'],
      dtype='object')
```

Step 4: Identifying Missing Values:

In [17]:
```
Missing_Values = netflix_data.isnull()  # 1. isnull(): Returns a boolean mask in
Missing_Values
```

Out[17]:

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score	Country Availability	Runtime	Director
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
9420	False	False	False	False	False	False	False	False	False
9421	False	False	False	False	False	False	False	False	False
9422	False	False	False	False	False	False	False	False	False
9423	False	False	False	False	False	False	False	False	False
9424	False	False	False	False	False	False	False	False	True

9425 rows × 29 columns

Counting Missing Values:

In [19]:
```
print("\nMissing Values for Each Column:\n")
Missing_Count = Missing_Values.sum()   # 1. isnull().sum(): Returns the count of
Missing_Count
```

```
Missing Values for Each Column:
```

Out[19]:
```
Title                       0
Genre                      25
Tags                       36
Languages                 170
Series or Movie             0
Hidden Gem Score           10
Country Availability       11
Runtime                     1
Director                 2305
Writer                   1810
Actors                    111
View Rating              2598
IMDb Score                  8
Rotten Tomatoes Score    3980
Metacritic Score         5343
Awards Received          4199
Awards Nominated For     3049
Boxoffice                5671
Release Date              208
Netflix Release Date        0
Production House         5032
Netflix Link                0
IMDb Link                 324
Summary                     5
IMDb Votes                 10
Image                       0
Poster                    938
TMDb Trailer                0
Trailer Site                1
dtype: int64
```

Step 5: Handling Missing Values

In [22]:
```
  # Handle missing values by dropping rows or filling with a specific value
cleaned_data = netflix_data.dropna()  # Drop rows with missing values
cleaned_data
```

Out[22]:

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Comedy Programmes,Romantic TV Comedies,Horror ...	Swedish, Spanish	Series	4.3
9	Joker	Crime, Drama, Thriller	Dark Comedies,Crime Comedies,Dramas,Comedies,C...	English	Movie	3.5
10	I	Action, Adventure, Fantasy, Sci-Fi	Dramas,Swedish Movies	English, Sanskrit	Movie	2.8
11	Harrys Daughters	Adventure, Drama, Fantasy, Mystery	Dramas,Swedish Movies	English	Movie	4.4
17	The Closet	Comedy	Korean Movies,Horror Movies,Mysteries	French	Movie	3.8
...	...	...	...	...	...	...
9411	50 First Dates	Comedy, Drama, Romance	Romantic Favourites,Romantic Comedies,Comedies...	English, Hawaiian, Mandarin	Movie	2.7
9412	21	Crime, Drama, History, Thriller	Dramas,Dramas based on a book,Police Dramas,Po...	English	Movie	2.5
9414	One Chance	Biography, Comedy, Drama, Music	Dramas,Biographical Dramas,Dramas based on rea...	English, Italian	Movie	3.0
9415	The Twilight Saga: Breaking Dawn: Part 1	Adventure, Drama, Fantasy, Romance, Thriller	Dramas,Romantic Dramas,Dramas based on a book,...	English, Portuguese	Movie	2.0

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score
9416	One for the Money	Action, Comedy, Crime, Thriller	Romantic Comedies,Action Comedies,Comedies,Pol...	English	Movie	1.3

2155 rows × 29 columns

Step 6: Summary Statistics of Numerical Columns:

Pandas describe() Method

In [33]:
```
numerical_summary = netflix_data.describe()
print("\nSummary Statistics for Numerical Columns:\n")
numerical_summary  # Generates summary statistics (mean, std, min, 25%, 50%, 75%
```

```
Summary Statistics for Numerical Columns:
```

Out[33]:

	Hidden Gem Score	IMDb Score	Rotten Tomatoes Score	Metacritic Score	Awards Received	Awards Nominated For	
count	9415.000000	9417.000000	5445.000000	4082.000000	5226.000000	6376.000000	9
mean	5.540733	6.955517	64.691276	58.113425	9.735936	16.035602	6
std	2.447462	0.899681	25.269466	17.143187	19.524116	32.209094	1
min	0.600000	1.600000	0.000000	6.000000	1.000000	1.000000	5
25%	3.400000	6.500000	49.000000	46.000000	1.250000	2.000000	9
50%	5.300000	7.000000	70.000000	59.000000	4.000000	6.000000	6
75%	8.100000	7.500000	85.000000	71.000000	9.000000	15.000000	5
max	9.800000	9.700000	100.000000	100.000000	300.000000	386.000000	2

Step 7: Explore Categorical Variables:

Pandas select_dtypes Method

In [37]:
```
    # Identify categorical columns
categorical_cols = netflix_data.select_dtypes(include=['object']).columns
categorical_cols

     #  In this code:
     #  1.  We first import the pandas library.
```

```
#  2.  We create a sample DataFrame `netflix_data` with columns.
#  3.  We use `select_dtypes()` to identify columns with categorical data
#  4.  We iterate through `categorical_cols` and for each column, we print
```

Out[37]:
```
Index(['Title', 'Genre', 'Tags', 'Languages', 'Series or Movie',
       'Country Availability', 'Runtime', 'Director', 'Writer', 'Actors',
       'View Rating', 'Boxoffice', 'Release Date', 'Netflix Release Date',
       'Production House', 'Netflix Link', 'IMDb Link', 'Summary', 'Image',
       'Poster', 'TMDb Trailer', 'Trailer Site'],
      dtype='object')
```

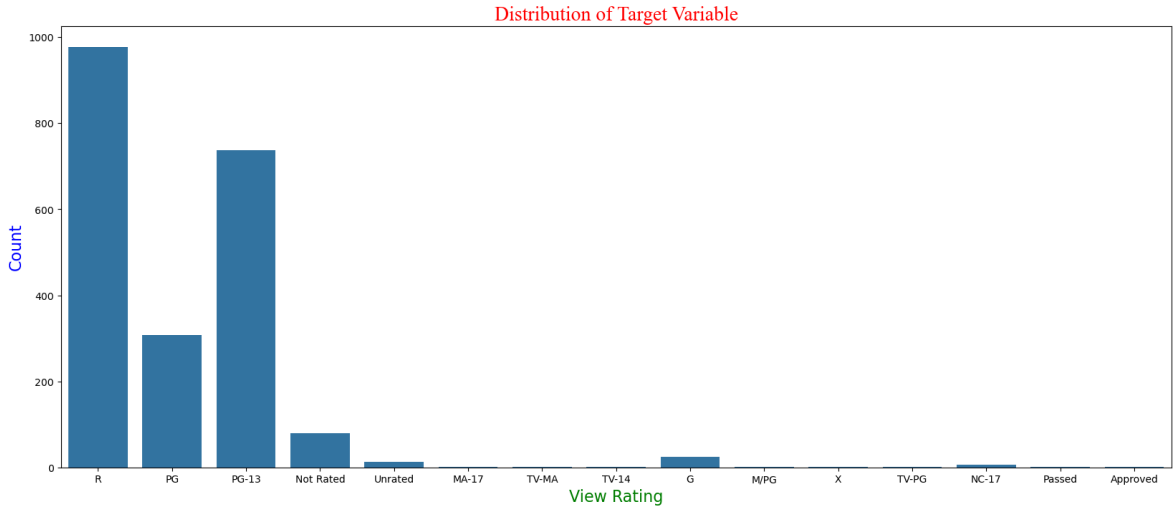
In [39]:
```
# Iterate through categorical columns and print unique values
for col in categorical_cols:
    print(f"{col}: {netflix_data[col].nunique()} unique values")
    print(netflix_data[col].unique()[:3], "\n")
    print()
    # The purpose of the loop is to explore the categorical variables by examini
    # This provides insights into:
    #    => Categorical distribution: Understanding how many unique categories e
    #    => Data diversity: Recognizing the variety of values within categorical
    # This exploration supports subsequent analysis involving these variables, s
```

```
Title: 9166 unique values
['Lets Fight Ghost' 'HOW TO BUILD A GIRL' 'The Con-Heartist']


Genre: 1531 unique values
['Crime, Drama, Fantasy, Horror, Romance' 'Comedy' 'Comedy, Romance']


Tags: 8552 unique values
['Comedy Programmes,Romantic TV Comedies,Horror Programmes,Thai TV Programmes'
 'Dramas,Comedies,Films Based on Books,British'
 'Romantic Comedies,Comedies,Romantic Films,Thai Comedies,Thai Films']


Languages: 1208 unique values
['Swedish, Spanish' 'English' 'Thai']


Series or Movie: 2 unique values
['Series' 'Movie']


Country Availability: 5281 unique values
['Thailand' 'Canada' 'Poland']


Runtime: 4 unique values
['< 30 minutes' '1-2 hour' '> 2 hrs']


Director: 4252 unique values
['Tomas Alfredson' 'Coky Giedroyc' 'Mez Tharatorn']


Writer: 6524 unique values
['John Ajvide Lindqvist' 'Caitlin Moran'
 'Pattaranad Bhiboonsawade, Mez Tharatorn, Thodsapon Thiptinnakorn']


Actors: 8835 unique values
['Lina Leandersson, Kåre Hedebrant, Per Ragnar, Henrik Dahl'
 'Cleo, Paddy Considine, Beanie Feldstein, Dónal Finn'
 'Kathaleeya McIntosh, Nadech Kugimiya, Pimchanok Leuwisetpaiboon, Thiti Mahayota
ruk']


View Rating: 24 unique values
['R' nan 'PG-13']


Boxoffice: 3589 unique values
['$2,122,065' '$70,632' nan]


Release Date: 4358 unique values
['12 Dec 2008' '08 May 2020' '03 Dec 2020']


Netflix Release Date: 1642 unique values
['2021-03-04' '2021-03-03' '2021-03-02']
```

```
Production House: 3276 unique values
['Canal+, Sandrew Metronome' 'Film 4, Monumental Pictures, Lionsgate' nan]


Netflix Link: 9425 unique values
['https://www.netflix.com/watch/81415947'
 'https://www.netflix.com/watch/81041267'
 'https://www.netflix.com/watch/81306155']


IMDb Link: 8826 unique values
['https://www.imdb.com/title/tt1139797'
 'https://www.imdb.com/title/tt4193072'
 'https://www.imdb.com/title/tt13393728']


Summary: 9415 unique values
['A med student with a supernatural gift tries to cash in on his abilities by fac
ing off against ghosts, till a wandering spirit brings romance instead.'
 'When nerdy Johanna moves to London, things get out of hand when she reinvents h
erself as a bad-mouthed music critic to save her poverty-stricken family.'
 'After her ex-boyfriend cons her out of a large sum of money, a former bank empl
oyee tricks a scam artist into helping her swindle him in retaliation.']


Image: 9425 unique values
['https://occ-0-4708-64.1.nflxso.net/dnm/api/v6/evlCitJPPCVCry0BZlEFb5-QjKc/AAAAB
cmgLCxN8dNahdY2kgd1hhcL2a6XrE92x24Bx5h6JFUvH5zMrv6lFWl_aWMt33b6DHvkgsUeDx_8Q1rmop
wT3fuF8Rq3S1hrkvFf3uzVv2sb3zrtU-LM1Zy1FfrAKD3nKNyA_RQWrmw.jpg?r=cd0'
 'https://occ-0-1081-999.1.nflxso.net/dnm/api/v6/evlCitJPPCVCry0BZlEFb5-QjKc/AAAA
Be_fxMSBM1E-sSoszr12SmkI-498sqBWrEyhkchdn4UklQVjdoPS_Hj-NhvgbePvwlDSzMTcrIE0kgiy-
zTEU_EaGg.jpg?r=35a'
 'https://occ-0-2188-64.1.nflxso.net/dnm/api/v6/evlCitJPPCVCry0BZlEFb5-QjKc/AAAAB
Sj6td_whxb4en62Ax5EKSKMl2lTzEK5CcBhwBdjRgF6SOJb4RtVoLhPAUWEskuOxPiaafxU1qauZDTJgu
wNQ9GstA.jpg?r=e76']


Poster: 8347 unique values
['https://m.media-amazon.com/images/M/MV5BOWM4NTY2NTMtZDZlZS00NTgyLWEzZDMtODE3ZGI
1MzI3ZmU5XkEyXkFqcGdeQXVyNzI1NzMxNzM@._V1_SX300.jpg'
 'https://m.media-amazon.com/images/M/MV5BZGUyN2ZlMjYtZTk2Yy00MWZiLWIyMDktMzFlMmE
zOWVlMGNiXkEyXkFqcGdeQXVyMTE1MzI2NzIz._V1_SX300.jpg'
 'https://m.media-amazon.com/images/M/MV5BODAzOGZmNjUtMTIyMC00NGU1LTg5MTMtZWY4MDd
iZjI0NGEwXkEyXkFqcGdeQXVyNzEyMTA5MTU@._V1_SX300.jpg']


TMDb Trailer: 9110 unique values
['https://www.youtube.com/watch?v=LqB6XJix-dM'
 'https://www.youtube.com/watch?v=eIbcxPy4okQ'
 'https://www.youtube.com/watch?v=md3CmFLGK6Y']


Trailer Site: 2 unique values
['YouTube' 'Vimeo' nan]
```

Step 8: Visualize the distribution of restaurant ratings

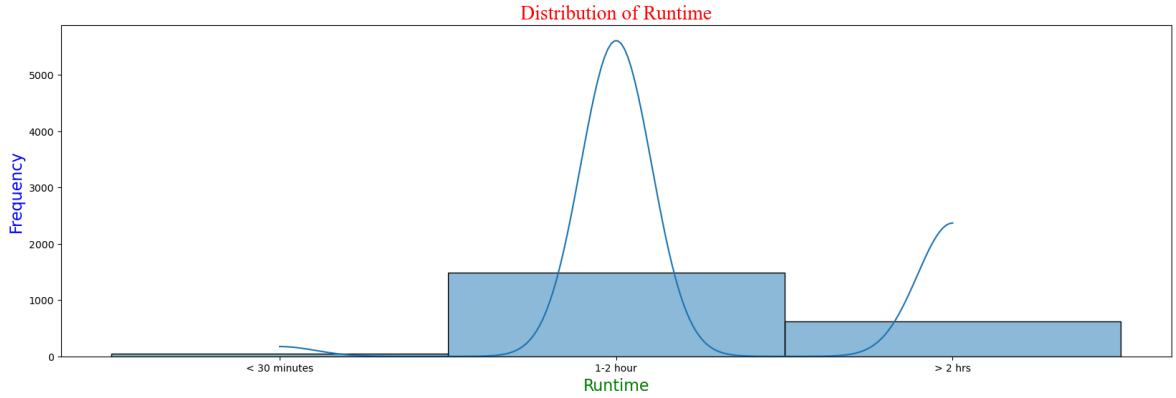
In [42]:
```
    # Create a count plot using Seaborn's countplot() function
plt.figure(figsize=(20,8))
sns.countplot(x='View Rating', data=cleaned_data)

    # Set title and labels
plt.title('Distribution of Target Variable', font='times new roman', color='red'
plt.xlabel('View Rating', color='green', fontsize=16)
plt.ylabel('Count',color='blue', fontsize=16)

    # Display the plot
plt.show()

    # In this example:

    # - We use Seaborn's countplot() function to create a count plot, which rep
    # - The x-axis represents the unique categories of the target variable.
    # - The y-axis represents the count of each category.

    # Information obtained from the plot:

    # - Class distribution: The plot shows the number of observations in each c
    # - Mode: The class with the highest frequency is the mode.
    # - Skewness: The plot reveals if the distribution is skewed or relatively
    # - Outliers: Unusual patterns or outliers may be visible.
```



Step 9: Visualize Distribution of Numerical Column

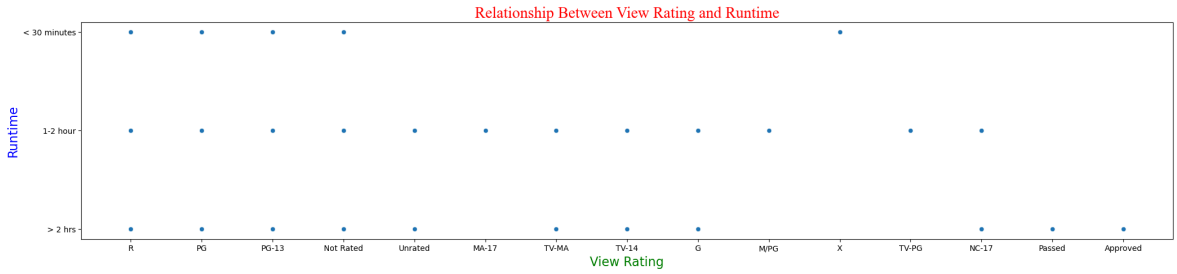
In [45]:
```
# Create a histogram with kernel density estimate (KDE)
plt.figure(figsize=(20, 6))
sns.histplot(cleaned_data['Runtime'], bins=20, kde=True)

# Set title and labels
plt.title('Distribution of Runtime', font='times new roman', color='red', fontsi
plt.xlabel('Runtime', color='green', fontsize=16)
plt.ylabel('Frequency',color='blue', fontsize=16)
```

```
# Show the plot
plt.show()
```

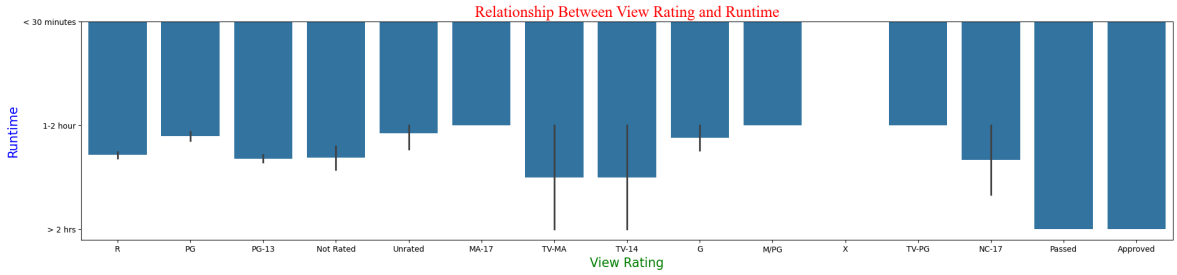


Step 10: Explore relationships between numerical columns

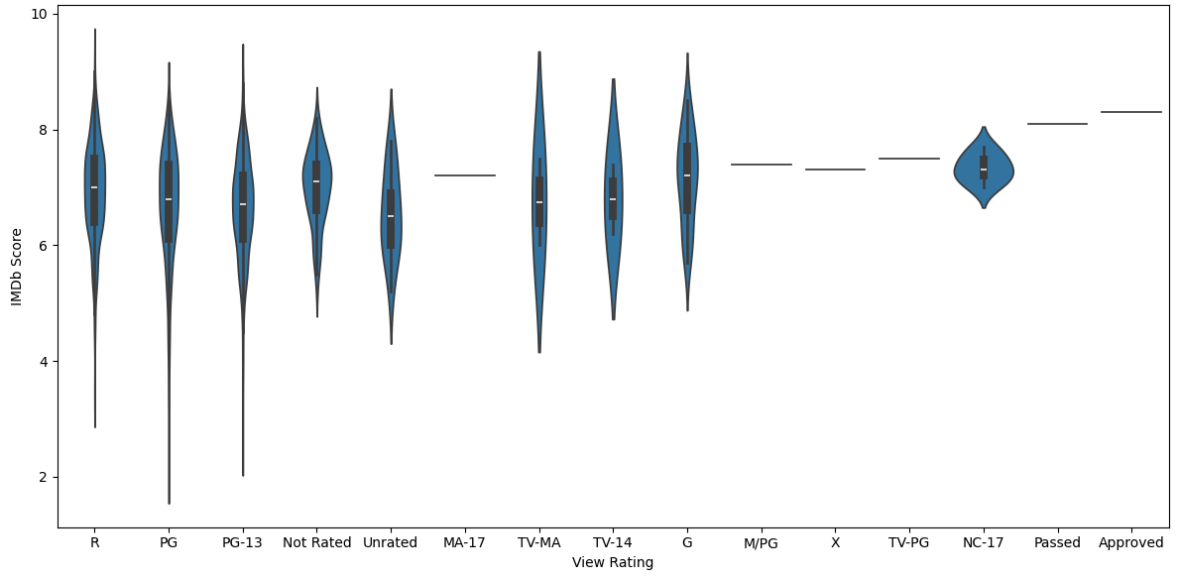
In [48]:
```
#  7 Explore relationships between numerical columns
plt.figure(figsize=(25, 5))
sns.scatterplot(x='View Rating', y='Runtime', data=cleaned_data)
plt.title("Relationship Between View Rating and Runtime", font='times new roman'
plt.xlabel("View Rating", color='green', fontsize=16)
plt.ylabel("Runtime", color='blue', fontsize=16)
plt.show()
```



In [50]:
```
#8: Introduce a different type of visualization
# 1. Bar Plot: Compares categorical data across groups.
# 2. Box Plot: Displays data distribution, median, and outliers.
# 3. Heatmap: Shows relationships between variables or data points.
# 4. Scatter Plot with Regression Line: Visualizes relationships between two var
# 5. Violin Plot: Combines box and density plots for data distribution.
# 6. Swarm Plot: Visualizes categorical data with overlapping points.
# 7. Pair Plot: Examines relationships between multiple variables.
# These visualizations help uncover patterns, relationships, and insights in dat

plt.figure(figsize=(25, 5))
sns.barplot(x='View Rating', y='Runtime', data=cleaned_data)
plt.title("Relationship Between View Rating and Runtime", font='times new roman'
plt.xlabel("View Rating", color='green', fontsize=16)
plt.ylabel("Runtime", color='blue', fontsize=16)
plt.show()
```



In [51]:
```
# 9: To display generated plots, use these Matplotlib functions:
  # 1. plt.show(): Displays the current figure.
  # 2. plt.display(): Displays plots in Jupyter notebooks
  # 3. plt.savefig('filename.png'): Saves plots to files.

# Additional functions for customization:
   # 1. plt.tight_layout(): Adjusts layout for better readability.
   # 2. plt.subplots_adjust(): Customizes spacing between subplots.
   # 3. plt.legend(): Adds legends to plots.

# Generate data
x = cleaned_data['View Rating']
y = cleaned_data['IMDb Score']

# Create plot
plt.figure(figsize=(12, 6))
sns.violinplot(x=x,y=y)


# Display plot
plt.tight_layout()
plt.show()
```



In [ ]:
```
#10: Based on the visualizations, here are some conclusions and recommendations

#Conclusions:

        #1. Content diversity: Netflix offers a diverse range of content, with

        #2. Genre popularity: Drama, Comedy, and Thriller are the most popular

        #3. Release year trend: The number of releases has increased significan
```

```
        #4. Rating distribution: Most content is rated TV-MA or TV-14.
        #5. Director and actor collaborations: Some directors and actors freque
#Recommendations:
        #1. Content strategy: Focus on producing more Drama, Comedy, and Thrill
        #2. Target audience: Cater to adult viewers (TV-MA/TV-14) and explore o
        #3. Original content: Increase original content production to maintain
        #4. Collaborations: Foster partnerships between successful directors an
        #5. Geographic expansion: Consider expanding into new markets, given th
#Interesting observations:
        #1. Correlation between ratings and reviews.
        #2. Differences in content preferences across regions.
        #3. Impact of awards on viewership.
#Future analysis suggestions:
        #1. Analyze user engagement metrics (e.g., watch time, completion rates
        #2. Investigate the effect of marketing campaigns on content popularity
        #3. Model predictive analysis for future content success.
#By exploring these insights, Netflix can refine its content strategy, improve
```