TASK-2

pip install pandas

import pandas as pd

# Reading a CSV file

df = pd.read\_csv('path\_to\_your\_file.csv')

# Display the first few rows of the dataframe

print(df.head())

# Display the dataframe info

print(df.info())

# Display summary statistics

print(df.describe())

**2. Performing Simple Data Cleaning Tasks**

**Handling Missing Values**

# Check for missing values

print(df.isnull().sum())

# Drop rows with any missing values

df\_dropped = df.dropna()

# Fill missing values with a specific value

df\_filled = df.fillna(0)

# Fill missing values with the mean of the column

df\_filled\_mean = df.fillna(df.mean())

# Check for duplicate rows

print(df.duplicated().sum())

# Drop duplicate rows

df\_no\_duplicates = df.drop\_duplicates()

### 3. Practicing Basic Data Manipulation Operations

#### Filtering Data

# Filter rows based on a condition

filtered\_df = df[df['column\_name'] > some\_value]

# Filter rows with multiple conditions

filtered\_df = df[(df['column\_name1'] > some\_value) & (df['column\_name2'] < another\_value)]

# Sort by a single column

sorted\_df = df.sort\_values(by='column\_name')

# Sort by multiple columns

sorted\_df = df.sort\_values(by=['column\_name1', 'column\_name2'], ascending=[True, False])

# Group by a single column and calculate the mean of each group

grouped\_df = df.groupby('column\_name').mean()

# Group by multiple columns and calculate the sum of each group

grouped\_df = df.groupby(['column\_name1', 'column\_name2']).sum()

import pandas as pd

# Reading the CSV file

df = pd.read\_csv('path\_to\_your\_file.csv')

# Viewing the first few rows and summary information

print(df.head())

print(df.info())

# Handling missing values

df = df.fillna(df.mean()) # Fill missing values with the mean

# Removing duplicates

df = df.drop\_duplicates()

# Filtering data

filtered\_df = df[df['some\_column'] > some\_value]

# Sorting data

sorted\_df = df.sort\_values(by='some\_column', ascending=False)

# Grouping data

grouped\_df = df.groupby('another\_column').sum()

# Display the results

print(filtered\_df.head())

print(sorted\_df.head())

print(grouped\_df.head())

### Improving Web Pages with CSS and Responsive Design

To enhance your web pages with CSS and implement responsive design techniques, follow these steps. This will ensure your website looks great and functions well across various devices and screen sizes.

#### Step 1: Add CSS Properties

Enhance your webpage by adding CSS properties to style various elements. Here's a basic example:

/\* CSS for styling elements \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #4CAF50;

color: white;

padding: 15px;

text-align: center;

}

nav {

margin: 10px 0;

text-align: center;

}

nav a {

margin: 0 15px;

text-decoration: none;

color: #333;

}

nav a:hover {

text-decoration: underline;

}

main {

padding: 20px;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px 0;

position: fixed;

bottom: 0;

width: 100%;

}

#### Step 2: Implement Responsive Design with Media Queries

Use media queries to adjust the layout and style of your webpage based on the screen size. Here's an example:

/\* Media queries for responsive design \*/

@media (max-width: 600px) {

header {

font-size: 18px;

}

nav a {

display: block;

margin: 10px 0;

}

main {

padding: 10px;

}

}

**Step 3: Test Responsiveness**

Test your website's responsiveness on different devices and browsers. Here are some tips:

1. **Use Browser DevTools**: Most modern browsers have developer tools that allow you to simulate different screen sizes and devices. For example, in Chrome, you can use the "Toggle Device Toolbar" to test responsiveness.
2. **Physical Devices**: Test on actual devices if possible, including phones, tablets, and different desktop monitors.
3. **Browser Testing Services**: Use services like BrowserStack or Sauce Labs to test your site across various browsers and devices.

**Step 4: Ensure Performance and Accessibility**

Prioritize performance and accessibility in your responsive design:

1. **Optimize Images**: Use responsive images with the **<img>** tag's **srcset** attribute or the **<picture>** element to serve appropriately sized images for different screen sizes.
2. **Minimize CSS and JavaScript**: Reduce file sizes by minifying CSS and JavaScript files.
3. **Accessibility**: Ensure your design is accessible by using semantic HTML elements, providing text alternatives for images, and ensuring sufficient color contrast.

**Example of Improved Web Page**

Here's a simple HTML structure incorporating the above CSS and media queries:

html

Copy code

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Responsive Web Page</title> <link rel="stylesheet" href="styles.css"> </head> <body> <header> <h1>My Responsive Web Page</h1> </header> <nav> <a href="#">Home</a> <a href="#">About</a> <a href="#">Contact</a> </nav> <main> <h2>Welcome!</h2> <p>This is a sample web page to demonstrate responsive design.</p> </main> <footer> &copy; 2024 My Website </footer> </body> </html>