Web technology 1 slips solution | WT and FDS CS-358 practical solution set for free

by **Daily cover** - September 30, 2024

TYBCS Web Technology 1 slip solutions

Introduction

In this article, we offer TYBCS Web Technology slip solutions, which make it possible to create interactive Web Technology 1 and Data Science slip solutions. In this set of useful slip solutions, we will examine a number of programs. These technologies consist of Python scripts, HTML, CSS, and PHP. You are guided through the use of these technologies by this solution set. Every practical activity is intended to improve your comprehension and give you practical experience as well as gain your knowledge, assisting you in developing your skills. This blog exists only to help you become a better coder and comprehend each technology concept. Web technology is an integral part of software and web development. The web technology practical slip solution set helps students develop their problem-solving skills. The practical slip solution focuses on real-world

Comments

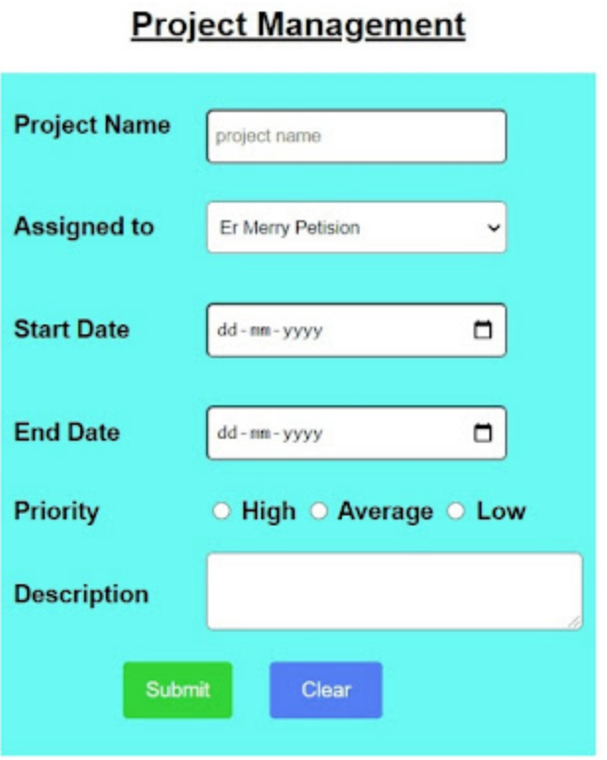
examples and problem-solving and enables learning to apply theoretical knowledge.

How to download TYBCS Web Technology 1 slip solutions

1. Go to the website visit the **dailycover.live** website, look for the article " **TYBCS Web Technology 1 slip solutions** "on the internet using a search label.go through the article find the download section and download it.
2. Telegram Groups: Joining Telegram groups devoted to more free material and sharing valuable material on this website. The Daily Cover website and telegram channel are popular platforms that both provide free-of-cost material.

Slip no -1

Q.1) Write the HTML code for generating the form as shown below. Apply the internal CSS to the following form to change the font size of the heading to 6pt and change the color to red and also change the background color to yellow.




```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Project Management Form</title>
    <style>
        body {
            background-color: yellow;
        }

        h1 {
            font-size: 6pt;
            color: red;
            text-align: center;
        }

        .form-container {
```

```
        background-color: #66CCCC;
        padding: 20px;
        width: 300px;
        margin: 0 auto;
        border-radius: 10px;
        font-family: Arial, sans-serif;
    }

    .form-container label {
        display: block;
        margin-bottom: 8px;
        font-weight: bold;
    }

    .form-container input,
    .form-container select,
    .form-container textarea {
        width: 100%;
        padding: 5px;
        margin-bottom: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    .form-container input[type="radio"] {
        width: auto;
    }

    .form-container button {
        width: 48%;
        padding: 10px;
        background-color: green;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    .form-container button.clear {
        background-color: blue;
    }

    .form-container button:hover {
        opacity: 0.8;
    }
  </style>
</head>
```

```
<body>

  <h1>Project Management</h1>

  <div class="form-container">
    <form>
      <label for="project-name">Project Name</label>
      <input type="text" id="project-name"
placeholder="project name">

      <label for="assigned-to">Assigned to</label>
      <select id="assigned-to">
        <option>Er Merry Petison</option>
        <!-- Add more options as needed -->
      </select>

      <label for="start-date">Start Date</label>
      <input type="date" id="start-date" placeholder="dd-
mm-yyyy">

      <label for="end-date">End Date</label>
      <input type="date" id="end-date" placeholder="dd-
mm-yyyy">

      <label for="priority">Priority</label>
      <input type="radio" id="high" name="priority"
value="high">
      <label for="high"
style="display:inline;">High</label>
      <input type="radio" id="average" name="priority"
value="average">
      <label for="average"
style="display:inline;">Average</label>
      <input type="radio" id="low" name="priority"
value="low">
      <label for="low" style="display:inline;">Low</label>

      <label for="description">Description</label>
      <textarea id="description" rows="4"
placeholder="Enter description"></textarea>

      <button type="submit">Submit</button>
      <button type="reset" class="clear">Clear</button>
    </form>
  </div>

</body>
```

```
</html>
```

Q.2 A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Iris dataset
# Assuming 'iris.csv' is the file containing the dataset
df = pd.read_csv('iris.csv')

# Count the frequency of each species
species_count = df['species'].value_counts()

# Create a pie plot
plt.figure(figsize=(6, 6))
plt.pie(species_count, labels=species_count.index,
autopct='%1.1f%%', startangle=90, colors=['skyblue',
'lightgreen', 'lightcoral'])
plt.title('Frequency of Iris Species')
plt.axis('equal')  # Ensures the pie chart is circular

# Display the pie chart
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

```
import pandas as pd

# Load the dataset
# Assuming 'winequality-red.csv' is the file containing the
dataset
df = pd.read_csv('winequality-red.csv')

# View basic statistical details of the data
basic_stats = df.describe()

# Display the statistics
print(basic_stats)
```

Slip no - 2

Q.1) Create HTML5 page with following specifications [
i) Title should be about your City.
ii) Color the background by Pink color.
iii) Place your city name at the top of page in large text and in blue color. iv) Add names of the landmarks in your city, each in different color, style and font
v) Add any image at the bottom. (Use inline CSS to format the web page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My City - [Your City Name]</title>
</head>
<body style="background-color: pink;">

    <!-- City name at the top with large text and blue color -->
    <h1 style="text-align: center; color: blue; font-size: 50px;">[Your City Name]</h1>

    <!-- Landmarks in the city with different styles, colors, and fonts -->
    <p style="color: red; font-family: Arial; font-size: 24px;">Landmark 1: [Landmark Name 1]</p>
    <p style="color: green; font-family: 'Courier New'; font-size: 22px;">Landmark 2: [Landmark Name 2]</p>
    <p style="color: orange; font-family: 'Times New Roman'; font-size: 20px;">Landmark 3: [Landmark Name 3]</p>
    <p style="color: purple; font-family: Verdana; font-size: 26px;">Landmark 4: [Landmark Name 4]</p>

    <!-- Image at the bottom -->
    <img src="[Image URL or Path]" alt="City Image" style="display: block; margin: 20px auto; width: 300px; height: auto;">

</body>
</html>
```

Q.2 A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

```
import pandas as pd

# Load the dataset
df = pd.read_csv('Data.csv')

# Display the dataset before handling missing values
print("Dataset before handling missing values:")
print(df)

# Replace missing values in 'salary' column with the mean of the 'salary' column
df['salary'].fillna(df['salary'].mean(), inplace=True)

# Replace missing values in 'age' column with the mean of the 'age' column
df['age'].fillna(df['age'].mean(), inplace=True)

# Display the dataset after handling missing values
print("\nDataset after handling missing values:")
print(df)

# Optionally save the updated dataset to a new CSV file
df.to_csv('Data_Updated.csv', index=False)
```

Q.2 B) Write a Python program to generate a line plot of name Vs salary

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('Data.csv')

# Plot name vs salary
plt.figure(figsize=(8, 6))
plt.plot(df['name'], df['salary'], marker='o', color='b', linestyle='-', markersize=8)

# Set plot title and labels
```

```
plt.title('Name vs Salary', fontsize=16)
plt.xlabel('Name', fontsize=12)
plt.ylabel('Salary', fontsize=12)

# Rotate the names on x-axis for better readability
plt.xticks(rotation=45)

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

Q.2 C) Download the heights and weights dataset and load the dataset froma given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display shape of the dataset.

```
import pandas as pd

# Load the dataset from a CSV file
df = pd.read_csv('heights_weights.csv')  # Replace
'heights_weights.csv' with your actual file path

# Display the shape of the dataset
print(f"Shape of the dataset: {df.shape}")

# Print the first 10 rows
print("\nFirst 10 rows of the dataset:")
print(df.head(10))

# Print the last 10 rows
print("\nLast 10 rows of the dataset:")
print(df.tail(10))

# Print 20 random rows
print("\nRandom 20 rows of the dataset:")
print(df.sample(20))
```

Slip no - 3

Q.1) Write a program using html with following CSS specifications-

1 The background colour of the company name should be in green.
ii. The text colour of the company name should be red.
iii. The heading should be large –with font "comic sans ms"
iv. The description of the company should be displayed in blue color in a paragraph.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Company Profile</title>
   <style>
      /* CSS Styles */
      .company-name {
         background-color: green;  /* Background color for company name */
         color: red;             /* Text color for company name */
         font-family: 'Comic Sans MS'; /* Font style */
         font-size: 36px;        /* Large font size */
         padding: 10px;          /* Padding around the text */
         text-align: center;      /* Centering the text */
      }
      .description {
         color: blue;            /* Text color for description */
         font-size: 20px;        /* Font size for description */
         margin: 20px;           /* Margin around the paragraph */
         text-align: center;      /* Centering the description */
      }
   </style>
</head>
<body>

   <!-- Company Name -->
   <h1 class="company-name">Your Company Name</h1>

   <!-- Company Description -->
   <p class="description">Welcome to Your Company! We provide the best services to our customers.</p>

</body>
```

```
</html>
```

Q.2 A)Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the actual path to your CSV file

# Set the style of seaborn
sns.set(style="whitegrid")

# Create box plots for each feature
features = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
plt.figure(figsize=(15, 10))

for i, feature in enumerate(features, start=1):
    plt.subplot(2, 2, i)  # Create a 2x2 grid of subplots
    sns.boxplot(x='species', y=feature, data=df)
    plt.title(f'Box plot of {feature} by Species')
    plt.xlabel('Species')
    plt.ylabel(feature)

# Adjust layout
plt.tight_layout()
plt.show()
```

Q.2 B) Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

```
 import pandas as pd

# Load the Heights and Weights dataset
df = pd.read_csv('heights_weights.csv')  # Replace 'heights_weights.csv' with your actual file path
```

```
# Display basic statistical details of the data
basic_stats = df.describe()

# Print the statistics
print("Basic Statistical Details:")
print(basic_stats)
```

Slip no 4

Q.1)Write a HTML code, which generate the following output

List of Books			
Item No	Item Name	Price	
		Rs.	Paise
1	Programming in Python	500	50
2	Programming in Java	345	00

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>List of Books</title>
  <style>
    body {
      font-family: Arial, sans-serif; /* Set a clean font for
the page */
      margin: 20px; /* Add some margin around the
page */
    }
    h1 {
      text-align: center; /* Center the heading */
    }
    table {
      width: 50%; /* Set the width of the table */
      margin: 0 auto; /* Center the table on the page */
      border-collapse: collapse; /* Collapse borders */
    }
    th, td {
      border: 1px solid black; /* Add border to table cells
*/
      padding: 8px; /* Add padding to cells */
      text-align: center; /* Center the text in the cells */
    }
    th {
```

```
        background-color: #f2f2f2; /* Light gray
background for header */
    }
  </style>
</head>
<body>

  <h1>List of Books</h1>

  <table>
    <tr>
      <th>Item No</th>
      <th>Item Name</th>
      <th>Price</th>
      <th>Rs.</th>
      <th>Paise</th>
    </tr>
    <tr>
      <td>1</td>
      <td>Programming in Python</td>
      <td>500</td>
      <td>50</td>
      <td></td>
    </tr>
    <tr>
      <td>2</td>
      <td>Programming in Java</td>
      <td>345</td>
      <td>00</td>
      <td></td>
    </tr>
  </table>

</body>
</html>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
```

```
random_data = np.random.randint(1, 101, size=50)

plt.figure(figsize=(12, 10))

plt.subplot(2, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange',
edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

plt.tight_layout()
plt.show()
```

Q.2 B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

```
import pandas as pd

# Load the dataset
```

```
df = pd.read_csv('User_Data.csv')  # Replace with the actual path to your CSV file

# Print the shape of the dataset
print(f"Shape of the dataset: {df.shape}")

# Print the number of rows and columns
num_rows, num_columns = df.shape
print(f"Number of Rows: {num_rows}")
print(f"Number of Columns: {num_columns}")

# Print data types of each feature
print("\nData Types of Each Feature:")
print(df.dtypes)

# Print feature names
print("\nFeature Names:")
print(df.columns.tolist())

# Print description of the data
print("\nDescription of the Data:")
print(df.describe(include='all'))  # Include='all' to get statistics for all columns, including non-numeric
```

Slip no 5

Q.1) Create following Bootstrap Web Layout Design and change Title, add your personal information, educational information, job profile


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Your Custom Title</title>
```

```
 <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/css/bootstrap.min.css">
</head>
<body>
 <div class="container text-center">
  <h1>Your Custom Title</h1>
  <p>Resize this responsive page to see the effect!</p>
  <div class="row">
   <div class="col-md-4">
    <h3>Personal Information</h3>
    <p>Add your personal information here.</p>
   </div>
   <div class="col-md-4">
    <h3>Educational Information</h3>
    <p>Add your educational information here.</p>
   </div>
   <div class="col-md-4">
    <h3>Job Profile</h3>
    <p>Add your job profile information here.</p>
   </div>
  </div>
 </div>

 <!-- Bootstrap JS and dependencies -->
 <script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"></script>
 <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1
/dist/umd/popper.min.js"></script>
 <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/js/bootstrap.min.js"></script>
</body>
</html>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
# Required libraries
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
data = np.random.randint(1, 101, 50)
```

```
# Create a figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
plt.plot(data, color='blue', marker='o', linestyle='-',
label='Random Data')
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)
plt.legend()

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(50), data, color='red', label='Random
Data')
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)
plt.legend()

# Histogram
plt.subplot(2, 2, 3)
plt.hist(data, bins=10, color='green', edgecolor='black')
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(data, patch_artist=True,
boxprops=dict(facecolor='orange'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```

Q.2 B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

```
# Required libraries
import pandas as pd

# Read the CSV file (assuming the file 'User_Data.csv' is in the same directory)
data = pd.read_csv('User_Data.csv')

# Print the shape of the data (number of rows and columns)
print("Shape of the data (rows, columns):", data.shape)

# Print the number of rows and columns
print("Number of rows:", data.shape[0])
print("Number of columns:", data.shape[1])

# Print the data types of each column
print("\nData types of each column:")
print(data.dtypes)

# Print the feature (column) names
print("\nFeature names (columns):")
print(list(data.columns))

# Print the description of the data (descriptive statistics)
print("\nDescription of the data:")
print(data.describe())
```

Slip no 6

Q.1) Create following Bootstrap Web Layout Design and set Header background color Blue, add your College name, set Menu section background color green create menu About Us, In content section add college information, background color yellow, Footer section background color red, add address of college.


```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width,
initial-scale=1.0">
 <title>College Web Layout</title>
 <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
 <style>
  .header {
   background-color: blue;
   color: white;
   text-align: center;
   padding: 20px;
  }
  .menu {
   background-color: green;
   color: white;
   padding: 20px;
   height: 500px;
  }
  .content {
   background-color: yellow;
   padding: 20px;
   height: 500px;
  }
  .footer {
   background-color: red;
   color: white;
   text-align: center;
   padding: 20px;
```

```
    }
  </style>
</head>
<body>

  <!-- Header Section -->
  <header class="header">
    <h1>Your College Name</h1>
  </header>

  <!-- Main Section with Menu and Content -->
  <div class="container-fluid">
    <div class="row">
      <!-- Menu Section -->
      <nav class="col-md-3 menu">
        <ul class="nav flex-column">
          <li class="nav-item">
            <a class="nav-link text-white" href="#">About
Us</a>
          </li>
          <li class="nav-item">
            <a class="nav-link text-white"
href="#">Contact</a>
          </li>
        </ul>
      </nav>

      <!-- Content Section -->
      <div class="col-md-9 content">
        <p>College Information: Your college's description,
mission, and other relevant details go here.</p>
      </div>
    </div>
  </div>

  <!-- Footer Section -->
  <footer class="footer">
    <p>College Address: Your college's address goes here.
</p>
  </footer>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Q.2 A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

```
# Required libraries
import pandas as pd

# Read the CSV file (assuming the file 'Data.csv' is in the same directory)
data = pd.read_csv('Data.csv')

# Display the data before handling missing values
print("Data before handling missing values:")
print(data)

# Calculate the mean of 'salary' and 'age' columns
mean_salary = data['salary'].mean()
mean_age = data['age'].mean()

# Replace missing values in 'salary' and 'age' columns with their respective means
data['salary'].fillna(mean_salary, inplace=True)
data['age'].fillna(mean_age, inplace=True)

# Display the data after handling missing values
print("\nData after replacing missing values in 'salary' and 'age':")
print(data)

# Optionally, save the updated data back to a new CSV file
data.to_csv('Data_Cleaned.csv', index=False)
```

Q.2 B) Write a Python program to generate a line plot of name Vs salary

```
# Required libraries
import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file (assuming the file 'Data.csv' contains 'name' and 'salary' columns)
data = pd.read_csv('Data.csv')

# Create a line plot: name vs salary
plt.figure(figsize=(10, 6))
```

```
plt.plot(data['name'], data['salary'], marker='o', color='b',
linestyle='-', label='Salary')

# Add title and labels
plt.title('Name vs Salary', fontsize=16)
plt.xlabel('Name', fontsize=12)
plt.ylabel('Salary', fontsize=12)

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Add grid and legend
plt.grid(True)
plt.legend()

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

Slip no 8

Q.1) Design an HTML form to accept two strings from the user. Write a PHP script for the following.
a. Find whether the small string appears at the start of the large string.
b. Find the position of the small string in the big string.
c. Compare both the string for first n characters, also the comparison should not be case sensitive
{HTML FILE}

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>String Comparison Form</title>
</head>
<body>
    <h1>String Comparison Form</h1>
    <form action="process.php" method="post">
        <label for="largeString">Enter the Large String:
</label><br>
        <input type="text" id="largeString"
name="largeString" required><br><br>

        <label for="smallString">Enter the Small String:
</label><br>
```

```
        <input type="text" id="smallString"
name="smallString" required><br><br>

        <label for="n">Enter number of characters for
comparison:</label><br>
        <input type="number" id="n" name="n" min="1"
required><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

[PHP FILE]

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $largeString = $_POST['largeString'];
    $smallString = $_POST['smallString'];
    $n = (int)$_POST['n'];

    $startsWith = strpos($largeString, $smallString) === 0;
    $position = strpos($largeString, $smallString);
    $compareResult = strncasecmp($largeString,
$smallString, $n) === 0;

    echo "<h1>Results</h1>";
    echo "<p>Large String: <strong>$largeString</strong>
</p>";
    echo "<p>Small String: <strong>$smallString</strong>
</p>";
    echo "<p>Does the small string appear at the start of
the large string? " . ($startsWith ? "Yes" : "No") . "</p>";

    if ($position !== false) {
        echo "<p>The position of the small string in the large
string is: <strong>$position</strong></p>";
    } else {
        echo "<p>The small string does not appear in the
large string.</p>";
    }

    echo "<p>Do the first $n characters of both strings
match (case insensitive)? " . ($compareResult ? "Yes" :
"No") . "</p>";
```

```
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2) Write a program in python to perform following task : [15] Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) (Use winequality-red.csv)

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('winequality-red.csv')
features = df.columns[:-1]
scaler = StandardScaler()
standardized_data = scaler.fit_transform(df[features])
standardized_df = pd.DataFrame(standardized_data,
columns=features)

print("Original Data:")
print(df.head())

print("\nStandardized Data:")
print(standardized_df.head())

standardized_df.to_csv('winequality_red_standardized.cs
v', index=False)
```

Slip no 9

Q.1) Write a PHP script for the following: Design a form having a text box and a drop down list containing any 3 separators(e.g. #, |, %, @, ! or comma) accept a strings from the user and also a separator.
a. Split the string into separate words using the given separator.

b. Replace all the occurrences of separator in the given string with some other separator.
c. Find the last word in the given string.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>String Separator Form</title>
</head>
<body>
    <h1>String Separator Form</h1>
    <form action="process.php" method="post">
        <label for="inputString">Enter a String:</label><br>
        <input type="text" id="inputString"
name="inputString" required><br><br>

        <label for="separator">Select a Separator:</label>
<br>
        <select id="separator" name="separator">
            <option value="#">#</option>
            <option value="|">|</option>
            <option value="%">%</option>
        </select><br><br>

        <label for="newSeparator">Replace with Separator:
</label><br>
        <input type="text" id="newSeparator"
name="newSeparator" required><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

PHP file

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $inputString = $_POST['inputString'];
    $separator = $_POST['separator'];
    $newSeparator = $_POST['newSeparator'];

    $words = explode($separator, $inputString);
    $replacedString = str_replace($separator,
$newSeparator, $inputString);
    $lastWord = end($words);
```

```
    echo "<h1>Results</h1>";
    echo "<p>Original String:
<strong>$inputString</strong></p>";
    echo "<p>Words: <strong>" . implode(', ', $words) . "
</strong></p>";
    echo "<p>String with Replaced Separator:
<strong>$replacedString</strong></p>";
    echo "<p>Last Word: <strong>$lastWord</strong>
</p>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot. Apply appropriate color, labels and styling options.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure
plt.figure(figsize=(12, 6))

# Line Chart
plt.subplot(1, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(1, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
```

```
# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

```
import matplotlib.pyplot as plt

subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

plt.figure(figsize=(8, 8))
plt.pie(marks, labels=subjects, autopct='%1.1f%%',
startangle=140, colors=['gold', 'lightcoral', 'lightskyblue',
'lightgreen', 'lightpink'])
plt.title('Marks Distribution by Subject', fontsize=16)
plt.axis('equal')
plt.show()
```

Q.2 C) Write a program in python to perform following task (Use winequality-red.csv) [5] Import Dataset and do the followings: a) Describing the dataset b) Shape of the dataset c) Display first 3 rows from dataset

```
import pandas as pd

df = pd.read_csv('winequality-red.csv')

print("Describing the Dataset:")
print(df.describe())

print("\nShape of the Dataset:")
print(df.shape)

print("\nFirst 3 Rows of the Dataset:")
print(df.head(3))
```

Slip no 10

Q.1) Write a script to accept two integers(Use html form having 2 textboxes). Write a PHP script to,

a. Find mod of the two numbers.
b. Find the power of first number raised to the second.
c. Find the sum of first n numbers (considering first number as n)
d. Find the factorial of second number. (Write separate function for each of the above operations.)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Integer Operations</title>
</head>
<body>
    <h1>Integer Operations Form</h1>
    <form action="process.php" method="post">
        <label for="num1">Enter First Integer:</label><br>
        <input type="text" id="num1" name="num1" required>
<br><br>

        <label for="num2">Enter Second Integer:</label><br>
        <input type="text" id="num2" name="num2" required>
<br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

PHP FILE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $num1 = (int)$_POST['num1'];
    $num2 = (int)$_POST['num2'];

    function findMod($a, $b) {
        return $a % $b;
    }

    function findPower($a, $b) {
        return pow($a, $b);
    }
```

```
    function sumOfFirstNNumbers($n) {
        return ($n * ($n + 1)) / 2;
    }

    function factorial($n) {
        if ($n < 0) return "Undefined";
        return ($n == 0) ? 1 : $n * factorial($n - 1);
    }

    $mod = findMod($num1, $num2);
    $power = findPower($num1, $num2);
    $sum = sumOfFirstNNumbers($num1);
    $factorial = factorial($num2);

    echo "<h1>Results</h1>";
    echo "<p>Mod of $num1 and $num2:
<strong>$mod</strong></p>";
    echo "<p>$num1 raised to the power of $num2:
<strong>$power</strong></p>";
    echo "<p>Sum of first $num1 numbers:
<strong>$sum</strong></p>";
    echo "<p>Factorial of $num2:
<strong>$factorial</strong></p>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2 A) Write a python program to Display column-wise mean, and median for the SOCRHeightWeight dataset.

```
import pandas as pd

# Load the SOCRHeightWeight dataset
df = pd.read_csv('SOCRHeightWeight.csv')  # Replace
with the actual file path

# Calculate column-wise mean
mean_values = df.mean()

# Calculate column-wise median
median_values = df.median()

# Display the results
print("Column-wise Mean:")
print(mean_values)
```

```
print("\nColumn-wise Median:")
print(median_values)
```

Q.2 B) Write a python program to compute sum of Manhattan distance between all pairs of points.

```
import numpy as np

# Sample data points (you can modify these points or read from a file)
points = np.array([[1, 2], [3, 5], [4, 1], [8, 0]])

def manhattan_distance(point1, point2):
    return np.abs(point1[0] - point2[0]) + np.abs(point1[1] - point2[1])

def total_manhattan_distance(points):
    total_distance = 0
    num_points = len(points)

    for i in range(num_points):
        for j in range(i + 1, num_points):  # Avoid duplicate pairs
            total_distance += manhattan_distance(points[i], points[j])

    return total_distance

# Compute the sum of Manhattan distances
result = total_manhattan_distance(points)

# Display the result
print(f"Sum of Manhattan distances between all pairs of points: {result}")
```

Slip no 11

Q.1) Create a button with different style (Secondary, Primary, Success, Error, Info, Warning, Danger) using BootStrap.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Bootstrap Buttons</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/css/bootstrap.min.css">
</head>
<body>

<div class="container mt-5">
    <h1>Bootstrap Button Styles</h1>

    <button class="btn btn-
secondary">Secondary</button>
    <button class="btn btn-primary">Primary</button>
    <button class="btn btn-success">Success</button>
    <button class="btn btn-danger">Error</button>
    <button class="btn btn-info">Info</button>
    <button class="btn btn-warning">Warning</button>
    <button class="btn btn-light">Light</button>
    <button class="btn btn-dark">Dark</button>
</div>

<script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2
/dist/umd/popper.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/js/bootstrap.min.js"></script>
</body>
</html>
```

Q.2 A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the
path to your dataset file
```

```
# Count the frequency of each species
species_count = df['species'].value_counts()

# Create a pie plot
plt.figure(figsize=(8, 8))
plt.pie(species_count, labels=species_count.index,
autopct='%1.1f%%', startangle=90, colors=['skyblue',
'lightgreen', 'lightcoral'])
plt.title('Frequency of Iris Species', fontsize=16)
plt.axis('equal')  # Equal aspect ratio ensures that pie
chart is circular.

# Display the pie chart
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

```
import pandas as pd

# Load the Wine Quality dataset
df = pd.read_csv('winequality-red.csv')  # Replace with
your actual file path

# Display basic statistical details of the data
basic_stats = df.describe()

# Print the statistics
print("Basic Statistical Details:")
print(basic_stats)
```

Slip no 12

Q.1) Write a PHP script for the following: Design a form to accept two numbers from the user. Give options to choose the arithmetic operation (use radio buttons). Display the result on the next form. (Use the concept of function and default parameters. Use 'include' construct or require statement)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Arithmetic Operations</title>
</head>
<body>
    <h1>Arithmetic Operations Form</h1>
    <form action="result.php" method="post">
        <label for="num1">Enter First Number:</label><br>
        <input type="number" id="num1" name="num1"
required><br><br>

        <label for="num2">Enter Second Number:</label>
<br>
        <input type="number" id="num2" name="num2"
required><br><br>

        <label>Choose an Operation:</label><br>
        <input type="radio" id="add" name="operation"
value="add" checked>
        <label for="add">Add</label><br>
        <input type="radio" id="subtract" name="operation"
value="subtract">
        <label for="subtract">Subtract</label><br>
        <input type="radio" id="multiply" name="operation"
value="multiply">
        <label for="multiply">Multiply</label><br>
        <input type="radio" id="divide" name="operation"
value="divide">
        <label for="divide">Divide</label><br><br>

        <input type="submit" value="Calculate">
    </form>
</body>
</html>
```

PHP CODE

```
<?php
include 'functions.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $num1 = (float)$_POST['num1'];
    $num2 = (float)$_POST['num2'];
    $operation = $_POST['operation'];

    $result = calculate($num1, $num2, $operation);
    echo "<h1>Result</h1>";
```

```
    echo "<p>Result of $operation between $num1 and
$num2 is: <strong>$result</strong></p>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

PHP CODE

```
<?php
function calculate($num1, $num2, $operation = 'add') {
    switch ($operation) {
        case 'add':
            return $num1 + $num2;
        case 'subtract':
            return $num1 - $num2;
        case 'multiply':
            return $num1 * $num2;
        case 'divide':
            return $num2 != 0 ? $num1 / $num2 : 'Division by
zero error';
        default:
            return 'Invalid operation';
    }
}
?>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
```

```
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange',
edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Q.2 B) Write a Python program to create data frame containing column name, salary, department add 10 rows with some missing and duplicate values to the data frame. Also drop all null and empty values. Print the modified data frame.

```
import pandas as pd
import numpy as np

# Create a DataFrame with some missing and duplicate
values
data = {
```

```
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank',
'Grace', 'Hannah', 'Ivy', 'Ivy'],
    'Salary': [70000, 80000, np.nan, 50000, 60000, 90000,
np.nan, 80000, 70000, 60000],
    'Department': ['HR', 'IT', 'Finance', 'IT', 'HR', 'Finance', 'IT',
'HR', None, 'Finance']
}

df = pd.DataFrame(data)

# Display the original DataFrame
print("Original DataFrame:")
print(df)

# Drop all null and empty values
df_cleaned = df.dropna()

# Display the modified DataFrame
print("\nModified DataFrame (after dropping null
values):")
print(df_cleaned)
```

Slip no 13

Q.1) Write a PHP script to create a chess board using CSS on table cells.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Chess Board</title>
    <style>
        table {
            border-collapse: collapse;
            width: 400px; /* Adjust the width as needed */
            height: 400px; /* Adjust the height as needed */
        }
        td {
            width: 50px; /* Each cell width */
            height: 50px; /* Each cell height */
        }
        .black {
```

```
            background-color: black;
        }
        .white {
            background-color: white;
        }
    </style>
</head>
<body>
    <h1>Chess Board</h1>
    <table>
        <?php
        for ($row = 0; $row < 8; $row++) {
            echo "<tr>";
            for ($col = 0; $col < 8; $col++) {
                $class = ($row % 2 === 0) ? (($col % 2 === 0) ?
'white' : 'black') : (($col % 2 === 0) ? 'black' : 'white');
                echo "<td class='$class'></td>";
            }
            echo "</tr>";
        }
        ?>
    </table>
</body>
</html>
```

Q.2 A) Write a Python program to create a graph to find relationship between the petal length and petal width. (Use iris.csv dataset)

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the
actual path to your dataset file

# Create a scatter plot to show the relationship between
petal length and petal width
plt.figure(figsize=(10, 6))
plt.scatter(df['petal_length'], df['petal_width'], c='blue',
alpha=0.5, edgecolor='k')

# Add titles and labels
plt.title('Relationship between Petal Length and Petal
Width', fontsize=16)
```

```
plt.xlabel('Petal Length (cm)', fontsize=12)
plt.ylabel('Petal Width (cm)', fontsize=12)

# Display grid
plt.grid(True)

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

Q.2 B) Write a Python program to find the maximum and minimum value of a given flattened array

```
import numpy as np

# Create a flattened array
array = np.array([3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5])

# Find the maximum value
max_value = np.max(array)

# Find the minimum value
min_value = np.min(array)

# Display the results
print(f"Flattened Array: {array}")
print(f"Maximum Value: {max_value}")
print(f"Minimum Value: {min_value}")
```

Slip no 14

Q.1) Create a container add row inside it and add 3 columns inside row using BootStrap.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Bootstrap Container with Columns</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/css/bootstrap.min.css">
</head>
<body>
```

```
<div class="container mt-5">
  <h1 class="text-center">Bootstrap Container with Three Columns</h1>
  <div class="row">
    <div class="col-md-4">
      <div class="p-3 border bg-light">Column 1</div>
    </div>
    <div class="col-md-4">
      <div class="p-3 border bg-light">Column 2</div>
    </div>
    <div class="col-md-4">
      <div class="p-3 border bg-light">Column 3</div>
    </div>
  </div>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Q. 2 A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array.

```
import numpy as np

# Create a flattened array
data = np.array([10, 20, 30, 40, 50])

# Create weights for the array
weights = np.array([1, 2, 3, 4, 5])

# Compute the weighted average
weighted_average = np.average(data, weights=weights)

# Display the result
print(f"Flattened Array: {data}")
print(f"Weights: {weights}")
```

```
print(f"Weighted Average: {weighted_average}")
```

Q. 2 B) Write a Python program to view basic statistical details of the data (Use advertising.csv)

```
import pandas as pd

# Load the advertising dataset
df = pd.read_csv('advertising.csv')  # Replace with your actual file path

# Display basic statistical details of the data
basic_stats = df.describe()

# Print the statistics
print("Basic Statistical Details:")
print(basic_stats)
```

Slip no 15

Q.1) Design a form to accept string from the user and perform the following operations a. To select first 5 words from the string b. Convert the given string to lowercase and then to Title case. c. Pad the given string with “*” from left and right both the sides. d. Remove the leading whitespaces from the given string. e. Find the reverse of given string.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>String Operations Form</title>
</head>
<body>
    <h1>String Operations Form</h1>
    <form action="process.php" method="post">
        <label for="inputString">Enter a String:</label><br>
        <textarea id="inputString" name="inputString" required></textarea><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
```

```
</html>
```

PHP CODE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $inputString = trim($_POST['inputString']);

    // a. Select first 5 words
    $words = explode(' ', $inputString);
    $firstFiveWords = implode(' ', array_slice($words, 0, 5));

    // b. Convert to lowercase and title case
    $lowercaseString = strtolower($inputString);
    $titleCaseString = ucwords($lowercaseString);

    // c. Pad with "*"
    $paddedString = '*' . $inputString . '*';

    // d. Remove leading whitespaces
    $trimmedString = ltrim($inputString);

    // e. Find the reverse of the string
    $reversedString = strrev($inputString);

    // Display results
    echo "<h1>Results</h1>";
    echo "<p>Original String:
<strong>$inputString</strong></p>";
    echo "<p>First 5 Words:
<strong>$firstFiveWords</strong></p>";
    echo "<p>Lowercase:
<strong>$lowercaseString</strong></p>";
    echo "<p>Title Case: <strong>$titleCaseString</strong>
</p>";
    echo "<p>Padded String:
<strong>$paddedString</strong></p>";
    echo "<p>Trimmed String:
<strong>$trimmedString</strong></p>";
    echo "<p>Reversed String:
<strong>$reversedString</strong></p>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange',
edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)
```

```
# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart

```
import matplotlib.pyplot as plt

# Define subject names and corresponding marks
subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(marks, labels=subjects, autopct='%1.1f%%',
startangle=90, colors=['gold', 'lightcoral', 'lightskyblue',
'lightgreen', 'lightpink'])
plt.title('Marks Distribution by Subject', fontsize=16)
plt.axis('equal')  # Equal aspect ratio ensures that pie
chart is circular.

# Show the pie chart
plt.show()
```

Slip no 16

Q.1) Write a PHP script for the following: Design a form to accept the marks of 5 different subjects of a student, having serial number, subject name & marks out of 100. Display the result in the tabular format which will have total, percentage and grade. Use only 3 text boxes.(Use array of form parameters)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Student Marks Input</title>
</head>
<body>
    <h1>Enter Marks for 5 Subjects</h1>
```

```
    <form action="process.php" method="post">
        <label for="subjects">Enter Subjects (comma-
separated):</label><br>
        <input type="text" id="subjects" name="subjects"
required placeholder="e.g., Math, Science, English,
History, Art"><br><br>

        <label for="marks">Enter Marks (comma-separated):
</label><br>
        <input type="text" id="marks" name="marks" required
placeholder="e.g., 85, 90, 78, 88, 92"><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

PHP CODE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $subjects = explode(',', $_POST['subjects']);
    $marks = explode(',', $_POST['marks']);

    // Remove any extra whitespace
    $subjects = array_map('trim', $subjects);
    $marks = array_map('trim', $marks);

    // Validate that we have 5 subjects and marks
    if (count($subjects) != 5 || count($marks) != 5) {
        echo "Please enter exactly 5 subjects and 5 marks.";
        exit;
    }

    // Convert marks to integers
    $marks = array_map('intval', $marks);

    // Calculate total and percentage
    $total = array_sum($marks);
    $percentage = $total / 5;

    // Determine grade
    if ($percentage >= 90) {
        $grade = 'A';
    } elseif ($percentage >= 80) {
        $grade = 'B';
```

```
    } elseif ($percentage >= 70) {
        $grade = 'C';
    } elseif ($percentage >= 60) {
        $grade = 'D';
    } else {
        $grade = 'F';
    }

    // Display results in a table
    echo "<h1>Result</h1>";
    echo "<table border='1'>
            <tr>
              <th>Serial No</th>
              <th>Subject</th>
              <th>Marks</th>
            </tr>";

    for ($i = 0; $i < 5; $i++) {
        echo "<tr>
                <td>" . ($i + 1) . "</td>
                <td>" . htmlspecialchars($subjects[$i]) . "</td>
                <td>" . $marks[$i] . "</td>
              </tr>";
    }

    echo "<tr>
            <td colspan='2'>Total</td>
            <td>$total</td>
          </tr>
          <tr>
            <td colspan='2'>Percentage</td>
            <td>" . number_format($percentage, 2) . "%</td>
          </tr>
          <tr>
            <td colspan='2'>Grade</td>
            <td>$grade</td>
          </tr>
          </table>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2 A) Write a python program to create two lists, one representing subject names and the other representing

marks obtained in those subjects. Display the data in a pie chart and bar chart.

```
import matplotlib.pyplot as plt

# Define subject names and corresponding marks
subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

# Create a pie chart
plt.figure(figsize=(12, 6))

# Pie Chart
plt.subplot(1, 2, 1)
plt.pie(marks, labels=subjects, autopct='%1.1f%%',
startangle=90, colors=['gold', 'lightcoral', 'lightskyblue',
'lightgreen', 'lightpink'])
plt.title('Marks Distribution by Subject', fontsize=16)
plt.axis('equal')  # Equal aspect ratio ensures that pie
chart is circular.

# Bar Chart
plt.subplot(1, 2, 2)
plt.bar(subjects, marks, color='skyblue',
edgecolor='black')
plt.title('Marks by Subject', fontsize=16)
plt.xlabel('Subjects', fontsize=12)
plt.ylabel('Marks Obtained', fontsize=12)
plt.ylim(0, 100)  # Setting the limit for y-axis

# Adjust layout to prevent overlap
plt.tight_layout()

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

Q.2 B) Write a python program to create a data frame for students' information such as name, graduation percentage and age. Display average age of students, average of graduation percentage.

```
import pandas as pd

# Create a DataFrame with students' information
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
```

```
    'Graduation_Percentage': [85.5, 90.0, 78.0, 88.5, 92.0],
    'Age': [21, 22, 20, 23, 21]
}

df = pd.DataFrame(data)

# Display the DataFrame
print("Students Information:")
print(df)

# Calculate the average age of students
average_age = df['Age'].mean()

# Calculate the average graduation percentage
average_graduation_percentage =
df['Graduation_Percentage'].mean()

# Display the averages
print(f"\nAverage Age of Students: {average_age:.2f}")
print(f"Average Graduation Percentage:
{average_graduation_percentage:.2f}")
```

Slip no 17

Q.1) Write a PHP script to sort the following associative array :
array("Sagar"=>"31","Vicky"=>"41",'Leena"=>"39","Rames h"=>"40") in
a) ascending order sort by Value
b) ascending order sort by Key
c) descending order sorting by Value
d) descending order sorting by Key

```
<?php
// Define the associative array
$array = array("Sagar" => "31", "Vicky" => "41", "Leena" =>
"39", "Ramesh" => "40");

// a) Ascending order sort by Value
asort($array);
echo "<h2>Ascending Order by Value:</h2>";
foreach ($array as $key => $value) {
    echo "$key => $value<br>";
}

// b) Ascending order sort by Key
```

```
ksort($array);
echo "<h2>Ascending Order by Key:</h2>";
foreach ($array as $key => $value) {
    echo "$key => $value<br>";
}

// c) Descending order sorting by Value
arsort($array);
echo "<h2>Descending Order by Value:</h2>";
foreach ($array as $key => $value) {
    echo "$key => $value<br>";
}

// d) Descending order sorting by Key
krsort($array);
echo "<h2>Descending Order by Key:</h2>";
foreach ($array as $key => $value) {
    echo "$key => $value<br>";
}
?>
```

Q.2 A) Write a Python program to draw scatter plots to compare two features of the iris dataset

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the actual path to your dataset file

# Set the style of seaborn for better aesthetics
sns.set(style="whitegrid")

# Create scatter plots for different pairs of features
plt.figure(figsize=(12, 8))

# Scatter plot: Petal Length vs Petal Width
plt.subplot(2, 2, 1)
sns.scatterplot(data=df, x='petal_length', y='petal_width', hue='species', style='species', palette='deep')
plt.title('Petal Length vs Petal Width')

# Scatter plot: Sepal Length vs Sepal Width
plt.subplot(2, 2, 2)
```

```
sns.scatterplot(data=df, x='sepal_length', y='sepal_width',
hue='species', style='species', palette='deep')
plt.title('Sepal Length vs Sepal Width')

# Scatter plot: Petal Length vs Sepal Length
plt.subplot(2, 2, 3)
sns.scatterplot(data=df, x='petal_length', y='sepal_length',
hue='species', style='species', palette='deep')
plt.title('Petal Length vs Sepal Length')

# Scatter plot: Petal Width vs Sepal Width
plt.subplot(2, 2, 4)
sns.scatterplot(data=df, x='petal_width', y='sepal_width',
hue='species', style='species', palette='deep')
plt.title('Petal Width vs Sepal Width')

# Adjust layout to prevent overlap
plt.tight_layout()

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

Q.2 B) Write a Python program to create a data frame containing columns name, age , salary, department . Add 10 rows to the data frame. View the data frame.

```
import pandas as pd

# Create a DataFrame with students' information
data = {
   'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve',
         'Frank', 'Grace', 'Hannah', 'Ivy', 'Jack'],
   'Age': [25, 30, 22, 35, 29, 28, 32, 24, 27, 31],
   'Salary': [50000, 60000, 55000, 70000, 65000,
          58000, 62000, 72000, 48000, 52000],
   'Department': ['HR', 'IT', 'Finance', 'Marketing', 'HR',
             'IT', 'Finance', 'HR', 'Marketing', 'IT']
}

# Create the DataFrame
df = pd.DataFrame(data)

# Display the DataFrame
print("DataFrame:")
print(df)
```

Slip no 18

Q.1) Write a menu driven program to perform the following operations on an associative array a. Reverse the order of each element's key-value pair. b. Traverse the element in an array in random order. c. Convert the array elements into individual variables. d. Display the elements of an array along with key.

```php
<?php
// Define the associative array
$array = array("Sagar" => "31", "Vicky" => "41", "Leena" => "39", "Ramesh" => "40");

function reverseKeyValuePairs($array) {
    return array_flip($array);
}

function traverseRandomOrder($array) {
    $keys = array_keys($array);
    shuffle($keys);
    foreach ($keys as $key) {
        echo "$key => " . $array[$key] . "<br>";
    }
}

function convertToVariables($array) {
    foreach ($array as $key => $value) {
        $$key = $value; // Convert to variable
    }
}

function displayElements($array) {
    foreach ($array as $key => $value) {
        echo "$key => $value<br>";
    }
}

while (true) {
    echo "<h1>Menu</h1>";
    echo "1. Reverse the order of each element's key-value pair<br>";
    echo "2. Traverse the element in an array in random order<br>";
    echo "3. Convert the array elements into individual variables<br>";
```

```
    echo "4. Display the elements of an array along with
keys<br>";
    echo "5. Exit<br>";

    $choice = (int)readline("Choose an option (1-5): ");

    switch ($choice) {
        case 1:
            $reversed = reverseKeyValuePairs($array);
            echo "<h2>Reversed Key-Value Pairs:</h2>";
            print_r($reversed);
            echo "<br>";
            break;
        case 2:
            echo "<h2>Random Order Traversal:</h2>";
            traverseRandomOrder($array);
            echo "<br>";
            break;
        case 3:
            convertToVariables($array);
            echo "<h2>Variables Created:</h2>";
            foreach ($array as $key => $value) {
                echo "$key = " . $$key . "<br>";
            }
            echo "<br>";
            break;
        case 4:
            echo "<h2>Elements of Array with Keys:</h2>";
            displayElements($array);
            echo "<br>";
            break;
        case 5:
            exit("Exiting program.");
        default:
            echo "Invalid option. Please try again.<br>";
            break;
    }
}
?>
```

Q.2 A) Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

```
import pandas as pd
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the
actual path to your dataset file

# Set the style of seaborn
sns.set(style="whitegrid")

# Create a box plot for each feature
plt.figure(figsize=(12, 10))

# Box plot for Sepal Length
plt.subplot(2, 2, 1)
sns.boxplot(x='species', y='sepal_length', data=df)
plt.title('Sepal Length by Species')

# Box plot for Sepal Width
plt.subplot(2, 2, 2)
sns.boxplot(x='species', y='sepal_width', data=df)
plt.title('Sepal Width by Species')

# Box plot for Petal Length
plt.subplot(2, 2, 3)
sns.boxplot(x='species', y='petal_length', data=df)
plt.title('Petal Length by Species')

# Box plot for Petal Width
plt.subplot(2, 2, 4)
sns.boxplot(x='species', y='petal_width', data=df)
plt.title('Petal Width by Species')

# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Q.2 B) Use the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 5 rows and random 10 row

```
import pandas as pd

# Load the heights and weights dataset
df = pd.read_csv('heights_weights.csv')  # Replace with
the actual file path
```

```
# Print the first 5 rows of the DataFrame
print("First 5 Rows:")
print(df.head())

# Print the last 5 rows of the DataFrame
print("\nLast 5 Rows:")
print(df.tail())

# Print 10 random rows of the DataFrame
print("\nRandom 10 Rows:")
print(df.sample(10))
```

Slip no 19

Q.1)Write a PHP script to accept 2 strings from the user, the first string should be a sentence and second can be a word. a. Delete a small part from first string after accepting position and number of characters to remove. b. Insert the given small string in the given big string at specified position without removing any characters from the big string. c. Replace some characters/ words from given big string with the given small string at specified position.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>String Operations</title>
</head>
<body>
   <h1>String Operations Form</h1>
   <form action="process.php" method="post">
      <label for="bigString">Enter a Sentence:</label><br>
      <textarea id="bigString" name="bigString" required>
</textarea><br><br>

      <label for="smallString">Enter a Word:</label><br>
      <input type="text" id="smallString"
name="smallString" required><br><br>

      <label for="position">Enter Position (0-indexed):
</label><br>
```

```
    <input type="number" id="position" name="position"
required><br><br>

    <label for="numChars">Number of Characters to
Remove:</label><br>
    <input type="number" id="numChars"
name="numChars" required><br><br>

    <label for="replacePosition">Replace Position (0-
indexed):</label><br>
    <input type="number" id="replacePosition"
name="replacePosition" required><br><br>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

PHP CODE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $bigString = $_POST['bigString'];
  $smallString = $_POST['smallString'];
  $position = (int)$_POST['position'];
  $numChars = (int)$_POST['numChars'];
  $replacePosition = (int)$_POST['replacePosition'];

  // a. Delete a part from the big string
  $deletedString = substr($bigString, 0, $position) .
substr($bigString, $position + $numChars);

  // b. Insert the small string into the big string at the
specified position
  $insertedString = substr($bigString, 0, $position) .
$smallString . substr($bigString, $position);

  // c. Replace part of the big string with the small string
at the specified position
  $replacedString = substr($bigString, 0,
$replacePosition) . $smallString . substr($bigString,
$replacePosition + strlen($smallString));

  // Display results
  echo "<h1>Results</h1>";
```

```
    echo "<p>Original Sentence:
<strong>$bigString</strong></p>";
    echo "<p>Deleted String:
<strong>$deletedString</strong></p>";
    echo "<p>Inserted String:
<strong>$insertedString</strong></p>";
    echo "<p>Replaced String:
<strong>$replacedString</strong></p>";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2) Write a Python program [15] 1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe. 2. To print the shape, number of rows-columns, data types, feature names and the description of the data 3. To Add 5 rows with duplicate values and missing values. Add a column ‘remarks’ with empty values. Display the data.

```
import pandas as pd
import numpy as np

# 1. Create a DataFrame with columns 'name', 'age', and
'percentage'
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve',
             'Frank', 'Grace', 'Hannah', 'Ivy', 'Jack'],
    'Age': [23, 30, 22, 25, 29,
            31, 28, 24, 27, 26],
    'Percentage': [85.5, 90.0, 78.0, 88.5, 92.0,
                   81.0, 75.0, 88.0, 95.0, 82.0]
}

df = pd.DataFrame(data)

# View the DataFrame
print("Initial DataFrame:")
print(df)

# 2. Print shape, number of rows-columns, data types,
feature names, and description of the data
print("\nShape of the DataFrame:", df.shape)  # (number
of rows, number of columns)
```

```
print("Number of Rows:", df.shape[0])
print("Number of Columns:", df.shape[1])
print("\nData Types:")
print(df.dtypes)
print("\nFeature Names:", df.columns.tolist())
print("\nDescription of the Data:")
print(df.describe())

# 3. Add 5 rows with duplicate values and missing values
duplicate_data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Age': [23, 30, np.nan, 25, 29],  # Adding a missing value
for Charlie
    'Percentage': [85.5, 90.0, 78.0, 88.5, 92.0]
}

df_duplicates = pd.DataFrame(duplicate_data)
df = pd.concat([df, df_duplicates], ignore_index=True)

# Add a new column 'remarks' with empty values
df['Remarks'] = ''

# Display the updated DataFrame
print("\nUpdated DataFrame with Duplicates and Missing
Values:")
print(df)
```

Slip no 20

Q.1) Write a menu driven program to perform the following operations on associative arrays: a) Split an array into chunks b) Sort the array by values without changing the keys. c) Filter the even elements from an array

```
<?php
// Define the associative array
$array = array("a" => 10, "b" => 15, "c" => 20, "d" => 25, "e"
=> 30, "f" => 35, "g" => 40, "h" => 45, "i" => 50);

function splitArrayIntoChunks($array, $chunkSize) {
    return array_chunk($array, $chunkSize);
}

function sortArrayByValues($array) {
    asort($array);
```

```
    return $array;
}

function filterEvenElements($array) {
    return array_filter($array, function($value) {
        return $value % 2 === 0;
    });
}

while (true) {
    echo "<h1>Menu</h1>";
    echo "1. Split array into chunks<br>";
    echo "2. Sort array by values without changing
keys<br>";
    echo "3. Filter even elements from array<br>";
    echo "4. Exit<br>";

    $choice = (int)readline("Choose an option (1-4): ");

    switch ($choice) {
        case 1:
            $chunkSize = (int)readline("Enter chunk size: ");
            $chunks = splitArrayIntoChunks($array,
$chunkSize);
            echo "<h2>Array Chunks:</h2>";
            print_r($chunks);
            echo "<br>";
            break;
        case 2:
            $sortedArray = sortArrayByValues($array);
            echo "<h2>Sorted Array:</h2>";
            print_r($sortedArray);
            echo "<br>";
            break;
        case 3:
            $filteredArray = filterEvenElements($array);
            echo "<h2>Even Elements:</h2>";
            print_r($filteredArray);
            echo "<br>";
            break;
        case 4:
            exit("Exiting program.");
        default:
            echo "Invalid option. Please try again.<br>";
            break;
    }
}
```

?>

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange',
edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
```

```
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

# Adjust layout to prevent overlap
plt.tight_layout()

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

Q.2 B) Add two outliers to the above data and display the box plot.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Add two outliers
outliers = [200, 250]  # Example outliers
random_data_with_outliers = np.append(random_data,
outliers)

# Set up the figure
plt.figure(figsize=(8, 6))

# Box Plot
plt.boxplot(random_data_with_outliers, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot with Outliers', fontsize=14)
plt.ylabel('Value', fontsize=12)

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

Slip no 21

Q.1) Create an array of 15 high temperatures, approximating the weather for a spring month, then find the average high temp, the five warmest high temps Display the result on the browser.

```
<?php
```

```php
// Create an array of 15 high temperatures for a spring month
$highTemperatures = array(60, 62, 65, 68, 70, 72, 75, 78, 80, 82, 85, 87, 90, 92, 95);

// Calculate the average high temperature
$averageTemp = array_sum($highTemperatures) / count($highTemperatures);

// Sort the array to find the five warmest temperatures
sort($highTemperatures);
$warmestTemps = array_slice($highTemperatures, -5);

// Display the results
echo "<h1>Temperature Analysis for Spring Month</h1>";
echo "<p>Average High Temperature: <strong>" . number_format($averageTemp, 2) . "°F</strong></p>";
echo "<p>Five Warmest High Temperatures: <strong>" . implode(', ', $warmestTemps) . "°F</strong></p>";
?>
```

Q.2 A) Import dataset “iris.csv”. Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data.

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the actual path to your dataset file

# Count the frequency of each species
species_count = df['species'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
species_count.plot(kind='bar', color=['skyblue', 'lightgreen', 'lightcoral'])
plt.title('Frequency of Iris Species', fontsize=16)
plt.xlabel('Species', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis='y')

# Show the plot
```

```
plt.show()
```

Q.2 B)Write a Python program to create a histogram of the three species of the Iris data.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the actual path to your dataset file

# Set the style of seaborn for better aesthetics
sns.set(style="whitegrid")

# Create a histogram for each feature colored by species
plt.figure(figsize=(12, 8))

# Histogram for Sepal Length
plt.subplot(2, 2, 1)
sns.histplot(data=df, x='sepal_length', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Sepal Length Distribution by Species')

# Histogram for Sepal Width
plt.subplot(2, 2, 2)
sns.histplot(data=df, x='sepal_width', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Sepal Width Distribution by Species')

# Histogram for Petal Length
plt.subplot(2, 2, 3)
sns.histplot(data=df, x='petal_length', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Petal Length Distribution by Species')

# Histogram for Petal Width
plt.subplot(2, 2, 4)
sns.histplot(data=df, x='petal_width', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Petal Width Distribution by Species')

# Adjust layout to prevent overlap
plt.tight_layout()
```

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

Slip no 22

Q.1) Write a menu driven program to perform the following queue related operations a) Insert an element in queue b) Delete an element from queue c) Display the contents of queue

```
<?php
// Initialize an empty queue
$queue = [];

// Function to insert an element into the queue
function enqueue(&$queue, $element) {
    array_push($queue, $element);
}

// Function to delete an element from the queue
function dequeue(&$queue) {
    if (empty($queue)) {
        return "Queue is empty, cannot delete an element.";
    }
    return array_shift($queue);
}

// Function to display the contents of the queue
function displayQueue($queue) {
    if (empty($queue)) {
        return "Queue is empty.";
    }
    return implode(", ", $queue);
}

// Menu-driven program
while (true) {
    echo "<h1>Queue Operations Menu</h1>";
    echo "1. Insert an element into the queue<br>";
    echo "2. Delete an element from the queue<br>";
    echo "3. Display the contents of the queue<br>";
    echo "4. Exit<br>";

    $choice = (int)readline("Choose an option (1-4): ");

    switch ($choice) {
        case 1:
```

```
        $element = readline("Enter the element to insert: ");
        enqueue($queue, $element);
        echo "Element '$element' inserted into the queue.
<br><br>";
        break;
      case 2:
        $removedElement = dequeue($queue);
        echo $removedElement . "<br><br>";
        break;
      case 3:
        echo "Contents of the queue: " .
displayQueue($queue) . "<br><br>";
        break;
      case 4:
        exit("Exiting program.");
      default:
        echo "Invalid option. Please try again.<br><br>";
        break;
    }
}
?>
```

Q.2) Dataset Name: winequality-red.csv [15] Write a program in python to perform following tasks a. Rescaling: Normalised the dataset using MinMaxScaler class b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) c. Normalizing Data (rescale each observation to a length of 1 (a unit norm). For this, use the Normalizer class.)

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler,
StandardScaler, Normalizer

# Load the wine quality dataset
df = pd.read_csv('winequality-red.csv')  # Replace with
your actual file path

# Display the first few rows of the original dataset
print("Original Dataset:")
print(df.head())

# a. Rescaling: Normalize the dataset using
MinMaxScaler
```

```
minmax_scaler = MinMaxScaler()
normalized_data = minmax_scaler.fit_transform(df)
normalized_df = pd.DataFrame(normalized_data,
columns=df.columns)

print("\nNormalized Dataset using MinMaxScaler:")
print(normalized_df.head())

# b. Standardizing Data (transform into a standard
Gaussian distribution)
standard_scaler = StandardScaler()
standardized_data = standard_scaler.fit_transform(df)
standardized_df = pd.DataFrame(standardized_data,
columns=df.columns)

print("\nStandardized Dataset:")
print(standardized_df.head())

# c. Normalizing Data (rescale each observation to a
length of 1)
normalizer = Normalizer()
normalized_unit_data = normalizer.fit_transform(df)
normalized_unit_df =
pd.DataFrame(normalized_unit_data,
columns=df.columns)

print("\nNormalized Dataset with Unit Norm:")
print(normalized_unit_df.head())
```

Slip no 23

Q.1) Write a menu driven program to perform the following stack related operations: a) Insert an element in stack b) Delete an element from stack c) Display the contents of stack

```
<?php
// Initialize an empty stack
$stack = [];


function push(&$stack, $element) {
    array_push($stack, $element);
}

// Function to delete an element from the stack
function pop(&$stack) {
```

```
    if (empty($stack)) {
        return "Stack is empty, cannot delete an element.";
    }
    return array_pop($stack);
}

// Function to display the contents of the stack
function displayStack($stack) {
    if (empty($stack)) {
        return "Stack is empty.";
    }
    return implode(", ", $stack);
}

// Menu-driven program
while (true) {
    echo "<h1>Stack Operations Menu</h1>";
    echo "1. Insert an element into the stack<br>";
    echo "2. Delete an element from the stack<br>";
    echo "3. Display the contents of the stack<br>";
    echo "4. Exit<br>";

    $choice = (int)readline("Choose an option (1-4): ");

    switch ($choice) {
        case 1:
            $element = readline("Enter the element to insert: ");
            push($stack, $element);
            echo "Element '$element' inserted into the stack.
<br><br>";
            break;
        case 2:
            $removedElement = pop($stack);
            echo $removedElement . "<br><br>";
            break;
        case 3:
            echo "Contents of the stack: " .
displayStack($stack) . "<br><br>";
            break;
        case 4:
            exit("Exiting program.");
        default:
            echo "Invalid option. Please try again.<br><br>";
            break;
    }
}
?>
```

Q.2) Dataset Name: winequality-red.csv [15] Write a program in python to perform following task a. Rescaling: Normalised the dataset using MinMaxScaler class b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) c. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0)

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler,
StandardScaler, Binarizer

# Load the wine quality dataset
df = pd.read_csv('winequality-red.csv')  # Replace with
your actual file path

# Display the first few rows of the original dataset
print("Original Dataset:")
print(df.head())

# a. Rescaling: Normalize the dataset using
MinMaxScaler
minmax_scaler = MinMaxScaler()
normalized_data = minmax_scaler.fit_transform(df)
normalized_df = pd.DataFrame(normalized_data,
columns=df.columns)

print("\nNormalized Dataset using MinMaxScaler:")
print(normalized_df.head())

# b. Standardizing Data (transform into a standard
Gaussian distribution)
standard_scaler = StandardScaler()
standardized_data = standard_scaler.fit_transform(df)
standardized_df = pd.DataFrame(standardized_data,
columns=df.columns)

print("\nStandardized Dataset:")
print(standardized_df.head())

# c. Binarizing Data using Binarizer class
threshold = 0.5  # Define a threshold for binarization
binarizer = Binarizer(threshold=threshold)
binarized_data = binarizer.fit_transform(df)
```

```
binarized_df = pd.DataFrame(binarized_data,
columns=df.columns)

print("\nBinarized Dataset (using threshold =
{}):".format(threshold))
print(binarized_df.head())
```

Slip no 24

Q.1) Write a PHP program to read two file names from user and append content of first file into second file.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>File Append Program</title>
</head>
<body>
    <h1>Append Content of One File to Another</h1>
    <form action="process.php" method="post">
        <label for="file1">Enter the name of the first file (with
extension):</label><br>
        <input type="text" id="file1" name="file1" required>
<br><br>

        <label for="file2">Enter the name of the second file
(with extension):</label><br>
        <input type="text" id="file2" name="file2" required>
<br><br>

        <input type="submit" value="Append Content">
    </form>
</body>
</html>
```

PHP CODE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $file1 = $_POST['file1'];
    $file2 = $_POST['file2'];
```

```
    // Check if the first file exists
    if (!file_exists($file1)) {
        echo "The file <strong>$file1</strong> does not
exist.";
        exit;
    }

    // Read the content of the first file
    $content = file_get_contents($file1);

    // Append the content to the second file
    file_put_contents($file2, $content, FILE_APPEND |
LOCK_EX);

    // Display success message
    echo "Content of <strong>$file1</strong> has been
appended to <strong>$file2</strong>.";
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2 A) Import dataset “iris.csv”. Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data.

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the
actual path to your dataset file

# Count the frequency of each species
species_count = df['species'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
species_count.plot(kind='bar', color=['skyblue', 'lightgreen',
'lightcoral'])
plt.title('Frequency of Iris Species', fontsize=16)
plt.xlabel('Species', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xticks(rotation=0)  # Rotate x-axis labels for better
readability
plt.grid(axis='y')
```

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

Q.2 B) Write a Python program to create a histogram of the three species of the Iris data.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with the actual path to your dataset file

# Set the style of seaborn for better aesthetics
sns.set(style="whitegrid")

# Create a histogram for each feature colored by species
plt.figure(figsize=(12, 8))

# Histogram for Sepal Length
plt.subplot(2, 2, 1)
sns.histplot(data=df, x='sepal_length', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Sepal Length Distribution by Species')

# Histogram for Sepal Width
plt.subplot(2, 2, 2)
sns.histplot(data=df, x='sepal_width', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Sepal Width Distribution by Species')

# Histogram for Petal Length
plt.subplot(2, 2, 3)
sns.histplot(data=df, x='petal_length', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Petal Length Distribution by Species')

# Histogram for Petal Width
plt.subplot(2, 2, 4)
sns.histplot(data=df, x='petal_width', hue='species', bins=10, kde=True, alpha=0.7, multiple='stack')
plt.title('Petal Width Distribution by Species')

# Adjust layout to prevent overlap
plt.tight_layout()
```

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

Slip no 25

Q.1) Write a menu driven program to perform various file operations. Accept filename from user. a) Display type of file. b) Display last modification time of file c) Display the size of file d) Delete the file

```
<?php
// Function to display the type of file
function getFileType($filename) {
    if (file_exists($filename)) {
        return filetype($filename);
    } else {
        return "File does not exist.";
    }
}

// Function to display the last modification time of the file
function getLastModificationTime($filename) {
    if (file_exists($filename)) {
        return date("F d Y H:i:s.", filemtime($filename));
    } else {
        return "File does not exist.";
    }
}

// Function to display the size of the file
function getFileSize($filename) {
    if (file_exists($filename)) {
        return filesize($filename) . " bytes";
    } else {
        return "File does not exist.";
    }
}

// Function to delete the file
function deleteFile($filename) {
    if (file_exists($filename)) {
        unlink($filename);
        return "File '$filename' has been deleted.";
    } else {
```

```
        return "File does not exist.";
    }
}

// Menu-driven program
while (true) {
    echo "<h1>File Operations Menu</h1>";
    $filename = readline("Enter the filename (with
extension): ");

    echo "1. Display type of file<br>";
    echo "2. Display last modification time of file<br>";
    echo "3. Display the size of file<br>";
    echo "4. Delete the file<br>";
    echo "5. Exit<br>";

    $choice = (int)readline("Choose an option (1-5): ");

    switch ($choice) {
        case 1:
            echo "Type of file: " . getFileType($filename) . "<br>
<br>";
            break;
        case 2:
            echo "Last modification time: " .
getLastModificationTime($filename) . "<br><br>";
            break;
        case 3:
            echo "Size of file: " . getFileSize($filename) . "<br>
<br>";
            break;
        case 4:
            echo deleteFile($filename) . "<br><br>";
            break;
        case 5:
            exit("Exiting program.");
        default:
            echo "Invalid option. Please try again.<br><br>";
            break;
    }
}
?>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram

and box plot. Apply appropriate color, labels and styling options.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and 100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-', markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data, color='green', alpha=0.7)
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange', edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True, boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

# Adjust layout to prevent overlap
plt.tight_layout()
```

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

Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

```
import matplotlib.pyplot as plt

# Define subject names and corresponding marks
subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(marks, labels=subjects, autopct='%1.1f%%',
startangle=90, colors=['gold', 'lightcoral', 'lightskyblue',
'lightgreen', 'lightpink'])
plt.title('Marks Distribution by Subject', fontsize=16)
plt.axis('equal')  # Equal aspect ratio ensures that pie
chart is circular.

# Show the pie chart
plt.show()
```

Slip no 26

Q.1)Consider the following entities and their relationship. [15]
Doctor (doc_no, dname, address ,city ,area) Hospital (hosp_no, hname, hcity) Doctor-Hospital related with many-one relationship. Create a RDB in 3NF for above and solve the following. Using above database write a script in PHP to print the Doctor visiting to the Hospital in tabular format. Accept Hospital name from user

SQL

```
CREATE DATABASE hospital_management;

USE hospital_management;

-- Create the Hospital table
CREATE TABLE Hospital (
    hosp_no INT PRIMARY KEY,
    hname VARCHAR(100),
```

```
    hcity VARCHAR(100)
);

-- Create the Doctor table
CREATE TABLE Doctor (
    doc_no INT PRIMARY KEY,
    dname VARCHAR(100),
    address VARCHAR(255),
    city VARCHAR(100),
    area VARCHAR(100),
    hosp_no INT,  -- Foreign key referencing Hospital
    FOREIGN KEY (hosp_no) REFERENCES
Hospital(hosp_no)
);



-- Insert sample hospitals
INSERT INTO Hospital (hosp_no, hname, hcity) VALUES
(1, 'City Hospital', 'New York'),
(2, 'Community Health Center', 'Los Angeles'),
(3, 'General Medical Clinic', 'Chicago');

-- Insert sample doctors
INSERT INTO Doctor (doc_no, dname, address, city, area,
hosp_no) VALUES
(1, 'Dr. Smith', '123 Elm St', 'New York', 'Downtown', 1),
(2, 'Dr. Jones', '456 Oak St', 'New York', 'Uptown', 1),
(3, 'Dr. Brown', '789 Pine St', 'Los Angeles', 'Westside', 2),
(4, 'Dr. Taylor', '321 Maple Ave', 'Chicago', 'Northside', 3);
```

PHP CODE

```
<?php
// Database connection parameters
$host = 'localhost'; // Your database host
$user = 'root';      // Your database username
$password = '';      // Your database password
$dbname = 'hospital_management'; // Your database
name

// Create a connection
$conn = new mysqli($host, $user, $password, $dbname);

// Check connection
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
}

// Check if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $hospital_name = $_POST['hospital_name'];

    // Prepare SQL query
    $sql = "SELECT d.dname, d.address, d.city, d.area
            FROM Doctor d
            JOIN Hospital h ON d.hosp_no = h.hosp_no
            WHERE h.hname = ?";

    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $hospital_name);
    $stmt->execute();
    $result = $stmt->get_result();

    // Display results in a table
    echo "<h1>Doctors Visiting $hospital_name</h1>";
    echo "<table border='1'>
            <tr>
                <th>Doctor Name</th>
                <th>Address</th>
                <th>City</th>
                <th>Area</th>
            </tr>";

    while ($row = $result->fetch_assoc()) {
        echo "<tr>
                <td>{$row['dname']}</td>
                <td>{$row['address']}</td>
                <td>{$row['city']}</td>
                <td>{$row['area']}</td>
              </tr>";
    }
    echo "</table>";

    // Close statement
    $stmt->close();
}

// Close connection
$conn->close();
?>

<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Doctor Information</title>
</head>
<body>
  <h1>Find Doctors by Hospital</h1>
  <form action="" method="post">
    <label for="hospital_name">Enter Hospital Name:
</label>
    <input type="text" id="hospital_name"
name="hospital_name" required>
    <input type="submit" value="Search">
  </form>
</body>
</html>
```

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a random array of 50 integers between 1 and
100
random_data = np.random.randint(1, 101, size=50)

# Set up the figure and subplots
plt.figure(figsize=(12, 10))

# Line Chart
plt.subplot(2, 2, 1)
plt.plot(random_data, color='blue', marker='o', linestyle='-',
markersize=5)
plt.title('Line Chart', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.grid(True)

# Scatter Plot
plt.subplot(2, 2, 2)
plt.scatter(range(len(random_data)), random_data,
color='green', alpha=0.7)
```

```
plt.title('Scatter Plot', fontsize=14)
plt.xlabel('Index', fontsize=12)
plt.ylabel('Value', fontsize=12)

# Histogram
plt.subplot(2, 2, 3)
plt.hist(random_data, bins=10, color='orange',
edgecolor='black', alpha=0.7)
plt.title('Histogram', fontsize=14)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Box Plot
plt.subplot(2, 2, 4)
plt.boxplot(random_data, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title('Box Plot', fontsize=14)
plt.ylabel('Value', fontsize=12)

# Adjust layout to prevent overlap
plt.tight_layout()

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

2.Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in bar chart.

```
import matplotlib.pyplot as plt

# Define subject names and corresponding marks
subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(subjects, marks, color=['skyblue', 'lightgreen',
'lightcoral', 'lightskyblue', 'lightpink'])
plt.title('Marks Obtained in Different Subjects',
fontsize=16)
plt.xlabel('Subjects', fontsize=12)
plt.ylabel('Marks', fontsize=12)
plt.ylim(0, 100)  # Setting the limit for y-axis
plt.grid(axis='y')
```

```
# Show the bar chart
plt.show()
```

Slip no 27

Q.1) Write a PHP program to read two file names from user and copy the content of first file into second file.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>File Copy Program</title>
</head>
<body>
    <h1>Copy Content of One File to Another</h1>
    <form action="process.php" method="post">
        <label for="file1">Enter the name of the first file (with
extension):</label><br>
        <input type="text" id="file1" name="file1" required>
<br><br>

        <label for="file2">Enter the name of the second file
(with extension):</label><br>
        <input type="text" id="file2" name="file2" required>
<br><br>

        <input type="submit" value="Copy Content">
    </form>
</body>
</html>
```

PHP CODE

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $file1 = $_POST['file1'];
    $file2 = $_POST['file2'];

    // Check if the first file exists
    if (!file_exists($file1)) {
        echo "The file <strong>$file1</strong> does not
exist.";
        exit;
```

```
    }

    // Read the content of the first file
    $content = file_get_contents($file1);

    // Write the content to the second file
    if (file_put_contents($file2, $content) !== false) {
        echo "Content of <strong>$file1</strong> has been
copied to <strong>$file2</strong>.";
    } else {
        echo "Error copying content to
<strong>$file2</strong>.";
    }
} else {
    echo "<p>No data received.</p>";
}
?>
```

Q.2) Create a dataset data.csv having two categorical column (the country column, and the purchased column). [15] a. Apply OneHot coding on Country column. b. Apply Label encoding on purchased column

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder,
LabelEncoder

# Load the dataset
df = pd.read_csv('data.csv')  # Replace with your actual
file path if needed

# Display the original dataset
print("Original Dataset:")
print(df)

# a. Apply One-Hot Encoding on the Country column
one_hot_encoder = OneHotEncoder(sparse=False)
country_encoded =
one_hot_encoder.fit_transform(df[['country']])

# Convert the One-Hot Encoded array to a DataFrame
country_encoded_df = pd.DataFrame(country_encoded,
columns=one_hot_encoder.get_feature_names_out(['coun
try']))
```

```
# Concatenate the One-Hot Encoded DataFrame with the
original DataFrame
df_one_hot = pd.concat([df, country_encoded_df],
axis=1).drop('country', axis=1)

print("\nDataset after One-Hot Encoding on 'country':")
print(df_one_hot)

# b. Apply Label Encoding on the Purchased column
label_encoder = LabelEncoder()
df_one_hot['purchased'] =
label_encoder.fit_transform(df_one_hot['purchased'])

print("\nDataset after Label Encoding on 'purchased':")
print(df_one_hot)
```

Slip no 28

Q.1) Write a program to read a flat file “student.dat”, calculate the percentage and display the data from file in tabular format.(Student.dat file contains rollno, name, OS, WT, DS, Python, Java, CN)

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder,
LabelEncoder

# Load the dataset
df = pd.read_csv('data.csv')  # Replace with your actual
file path if needed

# Display the original dataset
print("Original Dataset:")
print(df)

# a. Apply One-Hot Encoding on the Country column
one_hot_encoder = OneHotEncoder(sparse=False)
country_encoded =
one_hot_encoder.fit_transform(df[['country']])

# Convert the One-Hot Encoded array to a DataFrame
country_encoded_df = pd.DataFrame(country_encoded,
columns=one_hot_encoder.get_feature_names_out(['coun
try']))
```

```
# Concatenate the One-Hot Encoded DataFrame with the
original DataFrame
df_one_hot = pd.concat([df, country_encoded_df],
axis=1).drop('country', axis=1)

print("\nDataset after One-Hot Encoding on 'country':")
print(df_one_hot)

# b. Apply Label Encoding on the Purchased column
label_encoder = LabelEncoder()
df_one_hot['purchased'] =
label_encoder.fit_transform(df_one_hot['purchased'])

print("\nDataset after Label Encoding on 'purchased':")
print(df_one_hot)
```

Q.2) Write a Python program [15]
1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe.
2. To print the shape, number of rows-columns, data types, feature names and the description of the data.
3. To view basic statistical details of the data.
4. To Add 5 rows with duplicate values and missing values. Add a column ‘remarks’ with empty values. Display the data.

```
import pandas as pd
import numpy as np

# 1. Create a DataFrame with columns 'name', 'age', and
'percentage'
data = {
   'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve',
          'Frank', 'Grace', 'Hannah', 'Ivy', 'Jack'],
   'age': [23, 30, 22, 25, 29,
         31, 28, 24, 27, 26],
   'percentage': [85.5, 90.0, 78.0, 88.5, 92.0,
                81.0, 75.0, 88.0, 95.0, 82.0]
}

df = pd.DataFrame(data)

# View the DataFrame
print("Initial DataFrame:")
print(df)
```

```
# 2. Print shape, number of rows-columns, data types, feature names, and description of the data
print("\nShape of the DataFrame:", df.shape)  # (number of rows, number of columns)
print("Number of Rows:", df.shape[0])
print("Number of Columns:", df.shape[1])
print("\nData Types:")
print(df.dtypes)
print("\nFeature Names:", df.columns.tolist())
print("\nDescription of the Data:")
print(df.describe())

# 3. View basic statistical details of the data
print("\nBasic Statistical Details:")
print(df.describe(include='all'))

# 4. Add 5 rows with duplicate values and missing values
duplicate_data = {
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'age': [23, 30, np.nan, 25, 29],  # Adding a missing value for Charlie
    'percentage': [85.5, 90.0, 78.0, 88.5, 92.0]
}

df_duplicates = pd.DataFrame(duplicate_data)
df = pd.concat([df, df_duplicates], ignore_index=True)

# Add a new column 'remarks' with empty values
df['remarks'] = ''

# Display the updated DataFrame
print("\nUpdated DataFrame with Duplicates and Missing Values:")
print(df)
```

Slip no 29

Q.1) Consider the following entities and their relationships [15] Event (eno , title , date) Committee (cno , name, head , from_time ,to_time , status) Event and Committee have many to many relationship. Write a php script to accept title of event and modify status committee as working.

SQL

```
CREATE DATABASE event_management;
```

```
USE event_management;

-- Create the Event table
CREATE TABLE Event (
    eno INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100),
    date DATE
);

-- Create the Committee table
CREATE TABLE Committee (
    cno INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    head VARCHAR(100),
    from_time TIME,
    to_time TIME,
    status VARCHAR(50)
);

-- Create the Event_Committee table to establish many-to-
many relationship
CREATE TABLE Event_Committee (
    eno INT,
    cno INT,
    FOREIGN KEY (eno) REFERENCES Event(eno),
    FOREIGN KEY (cno) REFERENCES Committee(cno),
    PRIMARY KEY (eno, cno)  -- Composite primary key
);
```

PHP

```
<?php
// Database connection parameters
$host = 'localhost'; // Your database host
$user = 'root';      // Your database username
$password = '';      // Your database password
$dbname = 'event_management'; // Your database name

// Create a connection
$conn = new mysqli($host, $user, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

```
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $event_title = $_POST['event_title'];

    // Prepare SQL query to get the event number (eno) for
the provided title
    $sql = "SELECT eno FROM Event WHERE title = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $event_title);
    $stmt->execute();
    $stmt->store_result();

    // Check if the event exists
    if ($stmt->num_rows > 0) {
        $stmt->bind_result($eno);
        $stmt->fetch();

        // Update status of the committee associated with
this event
        $update_sql = "UPDATE Committee c
                       JOIN Event_Committee ec ON c.cno =
ec.cno
                       SET c.status = 'working'
                       WHERE ec.eno = ?";
        $update_stmt = $conn->prepare($update_sql);
        $update_stmt->bind_param("i", $eno);

        if ($update_stmt->execute()) {
            echo "Status of the committees associated with
the event '$event_title' has been updated to 'working'.";
        } else {
            echo "Error updating status: " . $conn->error;
        }

        $update_stmt->close();
    } else {
        echo "No event found with the title '$event_title'.";
    }

    $stmt->close();
}

// Close the connection
$conn->close();
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Modify Committee Status</title>
</head>
<body>
    <h1>Modify Committee Status</h1>
    <form action="" method="post">
        <label for="event_title">Enter Event Title:</label>
        <input type="text" id="event_title" name="event_title"
required>
        <input type="submit" value="Update Status">
    </form>
</body>
</html>
```

Q.2) Create a dataset data.csv having two categorical column (the country column, and the purchased column). [15] 1. Apply OneHot coding on Country column. 2. Apply Label encoding on purchased column

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder,
LabelEncoder

# Load the dataset
df = pd.read_csv('data.csv')  # Replace with your actual
file path if needed

# Display the original dataset
print("Original Dataset:")
print(df)

# 1. Apply One-Hot Encoding on the Country column
one_hot_encoder = OneHotEncoder(sparse=False)
country_encoded =
one_hot_encoder.fit_transform(df[['country']])

# Convert the One-Hot Encoded array to a DataFrame
country_encoded_df = pd.DataFrame(country_encoded,
columns=one_hot_encoder.get_feature_names_out(['coun
try']))
```

```
# Concatenate the One-Hot Encoded DataFrame with the
original DataFrame
df_one_hot = pd.concat([df, country_encoded_df],
axis=1).drop('country', axis=1)

print("\nDataset after One-Hot Encoding on 'country':")
print(df_one_hot)

# 2. Apply Label Encoding on the Purchased column
label_encoder = LabelEncoder()
df_one_hot['purchased'] =
label_encoder.fit_transform(df_one_hot['purchased'])

print("\nDataset after Label Encoding on 'purchased':")
print(df_one_hot)
```

Slip no 30

Q.1) Consider the following entities and their relationships [15] Student (Stud_id,name,class) Competition (c_no,c_name,type) Relationship between student and competition is many-many with attribute rank and year. Create a RDB in 3NF for the above and solve the following. Using above database write a script in PHP to accept a competition name from user and display information of student who has secured 1st rank in that competition.

SQl

```
CREATE DATABASE competition_management;

USE competition_management;

-- Create the Student table
CREATE TABLE Student (
    Stud_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    class VARCHAR(50)
);

-- Create the Competition table
CREATE TABLE Competition (
    c_no INT PRIMARY KEY AUTO_INCREMENT,
    c_name VARCHAR(100),
    type VARCHAR(50)
);
```

```
-- Create the Student_Competition table to establish
many-to-many relationship
CREATE TABLE Student_Competition (
    Stud_id INT,
    c_no INT,
    rank INT,
    year INT,
    FOREIGN KEY (Stud_id) REFERENCES Student(Stud_id),
    FOREIGN KEY (c_no) REFERENCES Competition(c_no),
    PRIMARY KEY (Stud_id, c_no)  -- Composite primary
key
);
```

PHP

```
<?php
// Database connection parameters
$host = 'localhost'; // Your database host
$user = 'root';      // Your database username
$password = '';      // Your database password
$dbname = 'competition_management'; // Your database
name

// Create a connection
$conn = new mysqli($host, $user, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $competition_name = $_POST['competition_name'];

    // Prepare SQL query to get student details for 1st rank
in the provided competition
    $sql = "SELECT s.Stud_id, s.name, s.class
            FROM Student s
            JOIN Student_Competition sc ON s.Stud_id =
sc.Stud_id
            JOIN Competition c ON sc.c_no = c.c_no
            WHERE c.c_name = ? AND sc.rank = 1";

    $stmt = $conn->prepare($sql);
```

```
    $stmt->bind_param("s", $competition_name);
    $stmt->execute();
    $result = $stmt->get_result();

    // Display results in a table
    echo "<h1>Students Secured 1st Rank in
$competition_name</h1>";
    echo "<table border='1'>
            <tr>
              <th>Student ID</th>
              <th>Name</th>
              <th>Class</th>
            </tr>";

    while ($row = $result->fetch_assoc()) {
        echo "<tr>
                <td>{$row['Stud_id']}</td>
                <td>{$row['name']}</td>
                <td>{$row['class']}</td>
              </tr>";
    }
    echo "</table>";

    // Close statement
    $stmt->close();
}

// Close connection
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Get Students with 1st Rank</title>
</head>
<body>
    <h1>Find Students by Competition</h1>
    <form action="" method="post">
        <label for="competition_name">Enter Competition
Name:</label>
        <input type="text" id="competition_name"
name="competition_name" required>
        <input type="submit" value="Search">
```

You might like

</form>

Facebook Twitter

Post a Comment (0)

Previous Post Next Post

Q.2) Write python program to [15] a. Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply

About Us

Welcome to DailyCover.live! On this website, we share valuable computer science materials and Board(JEE, NEET, CET), including job and internship opportunities, free courses, handwritten notes, and valuable information technology content. To stay informed, be sure to follow our blog and connect with us on all social media platforms.