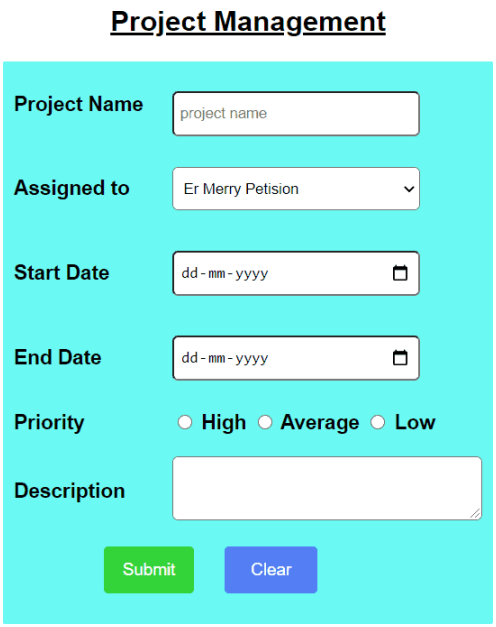
SLIP 1

Q.1) Write the HTML code for generating the form as shown below. Apply the internal CSS to 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.
[15]



SOLUTION:

```
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=utf-8" />

   <title>slip 1</title>
   <style>
      h2{
         font-size: 6pt;
         text-decoration: underline;
         color: red;
      }
```

```
        body{
            background-color: yellow;
        }
    </style>

</head>
<body>
    <h2>Project Management</h2>
    <br/>
    Project name<input type="text" name="name" id="name"/>
    <br />
    <br />
    <label for="ass">Assigned to</label>
    <select name="ass" id="Assigned to">
    <option value="Er Merry Petison">Er Merry Petison </option>
    <option value="Er Jerry Jetison">Er Jerry Jetison </option>
    <option value="Er Patrik Bateman">Er Patrik Bateman  </option>
    </select>
    <br />
    <br />
    Start date
    <input type="date" name="date" id="date"/>
    <br />
    <br />
    End date
    <input type="date" name="e" id="e"/>
```

```
    <br />
    <br />
    Priority
    <input type="radio" name="rad" id="rad" />High
    <input type="radio" name="avg" id="rad" />Average
    <input type="radio" name="low" id="rad" />Low

    <br />
    <br />
    Discription
    <textarea rows="2"></textarea>
    <br />
    <br />
    <input type="button" name="Submit" id="Submit" value="Submit"/>
    <input type="button" name="Reset" id="Reset" value="Reset"/>
</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) [10]

SOLUTION:

```
import pandas as pd
import matplotlib.pyplot as plt
# Load the iris dataset
iris = pd.read_csv("iris.csv")
# Get the counts of each species
```

```
species_counts = iris['Species'].value_counts()
# Plot the pie chart
fig, ax = plt.subplots()
ax.pie(species_counts, labels=species_counts.index)
ax.set_title("Species Proportions in Iris Dataset")
plt.show()
```

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

SOLUTION:

For entire data set

```
import pandas as pd
# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the dataset
print(iris.describe())
```

for a specific column from data set

```
import pandas as pd

# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the SepalLengthCm column
print(iris['SepalLengthCm'].describe())
```

SLIP 2

Q.1) Create HTML5 page with following specifications [15]

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)

SOLUTION:

```
<HTML>
<HEAD>
<TITLE>
<CENTER>My CITY
</CENTER>
</TITLE>
</HEAD>
<BODY BGCOLOR="pink">
<FONT SIZE="7" FACE="ARIAL" COLOR="BLUE">
<CENTER>PUNE</FONT><BR>
<BODY BGCOLOR="blue">
</CENTER>
<font size="6" face="arial" color="gray"<h1>VIMAN NAGAR
</h1>
</font><BR>
```

```
<font size="5" face="arial" color="yellow"<h2>SHANIVAR WADA
</h2>
</font>
<BR>
<font size="4" face="arial" color="red"<h3>KALYANI NAGAR</h3>
</font><BR>
<font size="3" face="arial" color="black"<h4>KP
</h4>
</font><BR>
<BODY BGCOLOR="PINK"<CENTER><MARQUEE BEHAVIOUR="SLIDE">
Wonderful Place To Visit</MARQUEE>
<MARQUEE BEHAVIOUR="SLIDE">Pune Is best City for Education</MARQUEE>
<textarea row=6 columns=6>Pune is a city. It is situated in maharashtra.It is a
hub for education.It is also known as an historical city.Pune is a city. It is
situated in maharashtra.It is a hub for education.It is also known as an historical
city.Pune is a city. It is situated in maharashtra.It is a hub for education.It is also
known as an historical city.
</textarea>
<IMG SRC="a.jpeg" WIDTH="400" HEIGHT="400" ALT="IMAGE CAN BE
DISPLAYED">
</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).

SOLUTION:

```
import pandas as pd
```

```
# Load the data dataset
data = pd.read_csv("data.csv")

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

# Replace the missing values in the salary and age columns with the mean of each column
data['salary'] = data['salary'].fillna(salary_mean)
data['age'] = data['age'].fillna(age_mean)
# View basic statistical details of the modified dataset
print(data.describe())
```

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

SOLUTION:

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

# Load the data dataset
data = pd.read_csv("data.csv")

# Replace the missing values in the salary column with the mean salary
salary_mean = data['salary'].mean()
data['salary'] = data['salary'].fillna(salary_mean)

# Generate a line plot of name vs salary
```

```
data.plot(x='name', y='salary', kind='line')
plt.title("Name vs Salary")
plt.xlabel("Name")
plt.ylabel("Salary")
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.

SOLUTION:

```
import pandas as pd

# Load the data dataset
data = pd.read_csv("Height and weights.csv")

# Print the first 10 rows of the dataset
print(data.head(10))

# Print the last 10 rows of the dataset
print(data.tail(10))

# Print 20 random rows of the dataset
print(data.sample(20))

# Display the shape of the dataset
print(data.shape)
```

SLIP 3

Q.1) Write a program using html with following CSS specifications-[15]

i. 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.

SOLUTION:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .company-name {
            background-color: green;
            color: red;
            font-family: 'Comic Sans MS', sans-serif;
            font-size: 36px;
        }
        .description {
            color: blue;
        }
    </style>
</head>
<body>
    <h1 class="company-name">Company Name</h1>
```

This is a description of the company. It is displayed in blue color in a paragraph.

</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)

SOLUTION:

```
import seaborn as sns

import matplotlib.pyplot as plt


# Load the iris dataset

iris = sns.load_dataset("iris")


# Create a figure with four subplots

fig, ax = plt.subplots(2, 2, figsize=(10, 8))


# Create a box plot of sepal length by species

sns.boxplot(x='species', y='sepal_length', data=iris, ax=ax[0, 0])

ax[0, 0].set_title("Sepal Length")


# Create a box plot of sepal width by species

sns.boxplot(x='species', y='sepal_width', data=iris, ax=ax[0, 1])

ax[0, 1].set_title("Sepal Width")


# Create a box plot of petal length by species
```

```
sns.boxplot(x='species', y='petal_length', data=iris, ax=ax[1, 0])
ax[1, 0].set_title("Petal Length")

# Create a box plot of petal width by species
sns.boxplot(x='species', y='petal_width', data=iris, ax=ax[1, 1])
ax[1, 1].set_title("Petal Width")

plt.show()
```

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

SOLUTION:

For entire data set

```
import pandas as pd
# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the dataset
print(iris.describe())
```

for a specific column from data set

```
import pandas as pd

# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the SepalLengthCm column
print(iris['SepalLengthCm'].describe())
```

SLIP 4

Q.1)Write a HTML code, which generate the following output [15]

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

SOLUTION:

```
<!DOCTYPE html>
<html>

<body>
   <table border="1">
      <tr>
         <th colspan="4"> LIST OF BOOKS
         </th>
      </tr>
      <tr>
         <th rowspan="2"> ITEM No</th>
         <th rowspan="2"> ITEM NAME</th>
         <th colspan="2">PRICE</th>
      </tr>
      <tr>
         <th>RS</th>
```

```
        <th>paise</th>
    </tr>
    <tr>
        <th>1</th>
        <td>PROGRAMMING IN PYTHON</td>
        <td>500</td>
        <td>50</td>
    </tr>
    <tr>
        <th>2</th>
        <td>PROGRAMMING IN JAVA</td>
        <td>385</td>
        <td>00</td>
    </tr>
    <tr>
        <th>3</th>
        <td>DATA STRUCTURES USING C </td>
        <td>285</td>
        <td>00</td>
    </tr>
    <tr>
        <th colspan="2"> TOTAL COST</th>
        <th>1131</th>
        <th>25</th>
    </tr>
</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.

SOLUTION:

```
import random

import matplotlib.pyplot as plt


# Generate a random array of 50 integers

data = [random.randint(0, 100) for _ in range(50)]


# Line chart

plt.plot(data, '-', color='red', label='Line chart')

plt.legend()

plt.show()


# Scatter plot

plt.scatter(range(50), data, color='green', label='Scatter plot')

plt.legend()

plt.show()


# Histogram

plt.hist(data, bins=10, color='blue', label='Histogram')

plt.legend()
```

```
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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)

SOLUTION:

```
        import pandas as pd

# Load the User_Data dataset
data = pd.read_csv("User_Data.csv")

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

# Print the feature names (column names)
```

```
print("Feature names:")
print(data.columns)

# Print a summary of the statistical properties of the dataset
print("Description:")
print(data.describe())
```

SLIP 5

Q.1) Create following Bootstrap Web Layout Design and change Title, add your personal information, educational information, job profile. [15]

SOLUTION:

```
<html lang="en">


<head>
   <title>Bootstrap Example</title>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css
">
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script
>
   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"><
/script>
</head>
```

```
<body>

  <div class="jumbotron text-center">
    <h1>My First Bootstrap Page</h1>

    37
    <p>Resize this responsive page to see the effect!</p>
  </div>

  <div class="container">
    <div class="row">
      <div class="col-sm-4">
        <h3>Personal Information</h3>
        <p>Add your personal information..</p>
        <p>...</p>
      </div>
      <div class="col-sm-4">
        <h3>Educational Information</h3>
        <p>Add your educational information....</p>
        <p>...</p>
      </div>
      <div class="col-sm-4">
        <h3>Job Profile</h3>
        <p>Add your job profile information.....</p>
        <p>...</p>
```

```
        </div>
      </div>
    </div>
</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.

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
```

```
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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)

SOLUTION:

```
import pandas as pd

# Load the User_Data dataset
data = pd.read_csv("User_Data.csv")

# Print the shape of the dataset (number of rows and columns)
print("Shape:", 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("Data types:")
```

```
print(data.dtypes)

# Print the feature names (column names)
print("Feature names:")
print(data.columns)

# Print a summary of the statistical properties of the dataset
print("Description:")
print(data.describe())
```

SLIP 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. [15]

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).

SOLUTION:

```
import pandas as pd


# Load the data dataset

data = pd.read_csv("data.csv")


# Calculate the mean of the salary and age columns

salary_mean = data['salary'].mean()
```

```
age_mean = data['age'].mean()

# Replace the missing values in the salary and age columns with the mean of each column
data['salary'] = data['salary'].fillna(salary_mean)
data['age'] = data['age'].fillna(age_mean)
# View basic statistical details of the modified dataset
print(data.describe())
```

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

SOLUTION:

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

# Load the data dataset
data = pd.read_csv("data.csv")

# Replace the missing values in the salary column with the mean salary
salary_mean = data['salary'].mean()
data['salary'] = data['salary'].fillna(salary_mean)

# Generate a line plot of name vs salary
data.plot(x='name', y='salary', kind='line')
plt.title("Name vs Salary")
plt.xlabel("Name")
plt.ylabel("Salary")
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.

SOLUTION:

```
import pandas as pd


# Load the data dataset
data = pd.read_csv("Height and weights.csv")


# Print the first 10 rows of the dataset
print(data.head(10))


# Print the last 10 rows of the dataset
print(data.tail(10))


# Print 20 random rows of the dataset
print(data.sample(20))


# Display the shape of the dataset
print(data.shape)
```

SLIP 7

Q.1) Design HTML 5 Page Using CSS Which Displays the following Navigation Bar [15]

SOLUTION:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Assingment 1 Set C Q2</title>
    <style>
        .navbar{
            background-color:rgb(170, 170, 250)
            }
        .navbar ul{
            overflow: auto;/*to stop overload of bacround color*/
        }
        .navbar ul li{
            list-style: none;
            float: left;
        }
        .navbar ul li a{
            display: block;
```

```
        text-decoration: none;
        color: blue;
        padding: 10px 15px;
      }
      .navbar ul li a:hover{
        color: white;
        background-color: gray;
      }
    </style>
</head>
<body>
    <header>
      <nav class="navbar">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">Java</a></li>
          <li><a href="#">HTML</a></li>
          <li><a href="#">CSS</a></li>
        </ul>
      </nav>
    </header>
</body>
</html>
```

Q.2) Write a Python program to perform the following tasks :

a. Apply OneHot coding on Country column.

b. Apply Label encoding on purchased column

(Data.csv have two categorical column the country column, and the purchased column).

SOLUTION:

```
import pandas as pd

from sklearn.preprocessing import LabelEncoder


# Load the Data dataset

data = pd.read_csv("Data.csv")


# OneHot encode the Country column

one_hot = pd.get_dummies(data['Country'])


# Label encode the Purchased column

le = LabelEncoder()

le.fit(data['Purchased'])

label_encoded = le.transform(data['Purchased'])


# Add the encoded columns to the original dataset

data = pd.concat([data, one_hot, pd.DataFrame(label_encoded,
columns=['Purchased'])], axis=1)


# Drop the original Country and Purchased columns

data.drop(['Country', 'Purchased'], axis=1, inplace=True)
```

SLIP 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. [15]

SOLUTION:

Form.html

```
<!DOCTYPE html>
<html>


<body>
   <form name=f method=get action=as3setb1.php>
      <center>
         <table>
            <tr>
               <th colspan=2>
                  <h1>Choose Menu</h1>
               </th>
            </tr>
            <tr>
               <td>Enter the string 1:</td>
               <td><input type=text name=s1></td>
            </tr><br>
            <tr>
```

```
            <td>Enter the string 2:</td>
            <td><input type=text name=s2></td>
        </tr><br>
        <tr>
            <td><input type=radio name=op value=1></td>
            <td>a.Find whether the small string appears at the start of the large string. .</td>
        </tr><br>
        <tr>
            <td><input type=radio name=op value=2></td>
            <td>b.Find the position of the small string in the big string. </td>
        </tr><br>
        <tr>
            <td><input type=radio name=op value=3></td>
            <td>c. Compare both the string for first n characters, also the comparison should not be case
                sensitive. </td>
        </tr><br>
        <tr>
            <th colspan=2><input type=submit name=sb value=Submit></th>
        </tr>
    </table>
  </center>
</body>

</html>
```

As3setbq1

```
<?php
include("form.html");
$x=$_GET['s1'];
$y=$_GET['s2'];
$ch=$_GET['op'];
switch($ch)
{
   case 1:
      function appears_at_start($x, $y) {
         $substring = substr($y, 0, strlen($x));
         return strcmp($x, $substring) == 0;
      }
      if (appears_at_start($x, $y))
      {
         echo "The small string appears at the start of the large string.";
      }
      else {
         echo "The small string does not appear at the start of the large string.";
      }

   break;
   case 2:
      $position = strpos($x, $y);
   if ($position !== false) {
   // small string was found in large string
```

```
        echo "The small string was found at position $position in the large string.";
    }
    else {
    // small string was not found in large string
        echo "The small string was not found in the large string.";
    }
    break;
    case 3:
        function compare_strings($a, $b, $n) {
            $substring1 = substr($a, 0, $n);
            $substring2 = substr($b, 0, $n);

            return strcasecmp($substring1, $substring2);
        }
        $result = compare_strings($x, $y, 3);
if ($result == 0) {
  echo "The strings are equal for the first 3 characters.";
} else if ($result > 0) {
  echo "The first string is greater than the second for the first 3 characters.";
} else {
  echo "The first string is less than the second for the first 3 characters.";
}

    break;
}
?>
```

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

SOLUTION:

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

# Load the winequality-red dataset
data = pd.read_csv("winequality-red.csv")

# Separate the features (X) and the target (y)
X = data.drop('quality', axis=1)
y = data['quality']

# Standardize the features using StandardScaler
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X)

# Check the mean and standard deviation of the standardized features
print("Mean:", X_standardized.mean(axis=0))
print("Standard deviation:", X_standardized.std(axis=0))
```

SLIP 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.

Find the last word in the given string.

SOLUTION

Form.html

```
<!-- c. Find the last word in the given string(Use strrstr() function). -->

<!DOCTYPE html>

<html>


<body>

   <form method="post" action="as3setb2.php">

   <label for="string">Enter a string:</label><br>

   <input type="text" id="string" name="string"><br>

   <label for="separator">Choose a separator:</label><br>

   <select name="separator">

     <option value="#">#</option>

     <option value="!">!</option>

     <option value="%">%</option>

     <option value=",">,</option>

   </select><br><br>

   <input type=radio name=op value=1>a.Split the string.
```

```
    <br>
    <input type=radio name=op value=2>b.Replace separator.
    <select name="newsep">
      <option value="#">#</option>
      <option value="!">!</option>
      <option value="%">%</option>
      <option value=",">,</option>
    </select>
    <br>
    <input type=radio name=op value=3>c.Find the last word.
    <br>
    <input type=submit name=sb value=Submit>
</body>
</html>
```

Php file

```
<?php
include("slip9.html");
$string = $_POST['string'];
$separator = $_POST['separator'];
$newsep=$_POST['newsep'];
$ch = $_POST['op'];

switch($ch)
{
    case 1:
      $arr1=str_split($string);
```

```
    $array_string=explode(" ",$string);
    $sep_string=implode("$separator",$array_string);
    echo "Seperated string = $sep_string<br>";
  break;
  case 2:
    $arr1=str_split($string);
    $array_string=explode(" ",$string);
    $sep_string=implode("$separator",$array_string);
    echo "First Seperated string = $sep_string<br>";
    $sep_string2=implode("$newsep",$array_string);
    echo "String with new Seperator = $sep_string2<br>";
  break;
  case 3:
    $arr1=str_split($string);
    $array_string=explode(" ",$string);
    $sep_string=implode("$separator",$array_string);
    echo "Seperated string = $sep_string<br>";
    $last_word = strrchr($sep_string,"$separator");
    echo "last word = $last_word<br>";
  break;
  }
?>
```

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. [5]

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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. [5]

SOLUTION:

```
import matplotlib.pyplot as plt


# Create the lists of subject names and marks
subject_names = ['Math', 'Physics', 'Chemistry', 'Biology']
marks = [80, 90, 75, 85]


# Create the pie chart
plt.pie(marks, labels=subject_names)


# Add a title and show the plot
plt.title("Subject Marks")
plt.show()
```

C) Write a program in python to perform following task (Use winequality-red.csv) Import Dataset and do the followings:

a) Describing the dataset

b) Shape of the dataset

c) Display first 3 rows from dataset

SOLUTION:

```
import pandas as pd


# Load the winequality-red dataset
```

```
data = pd.read_csv("winequality-red.csv")

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

# Print a summary of the statistical properties of the dataset
print("Description:")
print(data.describe())

# Display the first 3 rows of the dataset
print("First 3 rows:")
print(data.head(3))
```

SLIP 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.) [15]

SOLUTION:

```
<html>
   <body>
      <form action="slip10.php" method="post">
         Enter first number
         <input type="number" name="t1"><br>
         Enter Second number
         <input type="number" name="t2"><br>
         <input type="submit" value="display">
      </form>
   </body>
</html>
<?php
   function mod($x,$y)
   {
      $z=$x%$y;
      echo "Mod of $x and $y = $z<br>";
   }
```

```
function power($x,$y)
{
   $f=1;
   $n1=$y;
   while($n1>0)
   {
      $f=$f*$x;
      $n1--;
   }
   echo "$x raised to powre to $y =$f<br>";
}
function fact($y)
{
   $i=1;$f=1;
   while($i<=$y)
   {
      $f=$f*$i;
      $i++;
   }
   echo "Factorrial of $y = $f";
}
function sum($y)
{
   $sum=0;
   $i=1;
   while($i<=$y){
```

```
        $sum=$sum+$i;
        $i++;
      }
      echo "Sum of first $y number = $sum<br>";
   }
   $x=$_POST['t1'];
   $y=$_POST['t2'];
   mod($x,$y);
   power($x,$y);
   sum($y);
   fact($y);
?>
```

Q.2 A) Write a python program to Display column-wise mean, and median for SOCR- HeightWeight dataset. [10]

SOLUTION:

```
import pandas as pd

# Load the SOCR-HeightWeight dataset
data = pd.read_csv("SOCR-HeightWeight.csv")

# Calculate the column-wise mean and median
mean = data.mean()
median = data.median()
```

```
# Print the mean and median for each column
print("Mean:")
print(mean)
print("\nMedian:")
print(median)
```

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

SOLUTION:

```
# Define a function to calculate the Manhattan distance between two points
def manhattan_distance(point1, point2):
    distance = 0
    for i in range(len(point1)):
        distance += abs(point1[i] - point2[i])
    return distance

# Define a list of points
points = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]

# Initialize the sum of distances to zero
sum_distances = 0

# Iterate over the points and calculate the sum of the Manhattan distances
for i in range(len(points)):
    for j in range(i+1, len(points)):
        sum_distances += manhattan_distance(points[i], points[j])
```

```
# Print the sum of distances
print("Sum of distances:", sum_distances)
```

SLIP 11

Q.1) Create a button with different style (Secondary, Primary, Success, Error, Info, Warning, Danger) using BootStrap. [15]

SOLUTION:

```
<!DOCTYPE html>

<html>

<head>

        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.c
ss" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7S
k" crossorigin="anonymous">

        <style>


        </style>

</head>

<body><button type="button" class="btn btn-primary">Primary</button>

<button type="button" class="btn btn-secondary">Secondary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-light">Light</button>

<button type="button" class="btn btn-dark">Dark</button>


<button type="button" class="btn btn-link">Link</button>

</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)

SOLUTION:

```
import pandas as pd
import matplotlib.pyplot as plt
# Load the iris dataset
iris = pd.read_csv("iris.csv")
# Get the counts of each species
species_counts = iris['Species'].value_counts()
# Plot the pie chart
fig, ax = plt.subplots()
ax.pie(species_counts, labels=species_counts.index)
ax.set_title("Species Proportions in Iris Dataset")
plt.show()
```

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

SOLUTION:

```
import pandas as pd
# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the dataset
print(iris.describe())
```

for a specific column from data set

```
import pandas as pd

# Load the iris dataset
iris = pd.read_csv("iris.csv")
# View basic statistical details of the SepalLengthCm column
print(iris['SepalLengthCm'].describe())
```

SLIP 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) [15]

SOLUTION:

```
<!DOCTYPE html>
<html>
<head>
   <title>Calculator</title>
</head>
<body>
   <form method="post" action="slip12.php">
      <label for="num1">Number 1:</label>
      <input type="text" name="num1" id="num1"><br>
      <label for="num2">Number 2:</label>
      <input type="text" name="num2" id="num2"><br>
      <label>Operation:</label><br>
      <input type="radio" name="operation" value="add" checked> Add<br>
      <input type="radio" name="operation" value="subtract"> Subtract<br>
      <input type="radio" name="operation" value="multiply"> Multiply<br>
      <input type="radio" name="operation" value="divide"> Divide<br>
      <input type="submit" name="submit" value="Calculate">
   </form>
</body>
</html>
```

```php
<?php

  $num1 = $_POST['num1'];
  $num2 = $_POST['num2'];
  $operation = $_POST['operation'];
//   include 'slip12.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 $num1 / $num2;
    default:
      return "Invalid operation";
  }
}

$result = calculate($num1, $num2, $operation);

echo "The result is: $result";
```

?>

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. [10]

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
```

```
plt.show()


# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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.

SOLUTION:

```
import pandas as pd


# Create the data frame
df = pd.DataFrame({'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Greta', 'Hannah', 'Igor', 'Jill'],
                   'Salary': [50000, 60000, 55000, None, 75000, None, 80000, 70000, None, 65000],
                   'Department': ['Sales', 'Marketing', 'Sales', 'HR', 'HR', 'Marketing', 'Sales', 'HR', 'HR', 'Sales']})


# Store the original data in a separate variable
original_df = df.copy()


# Print the original data
print("Original data:")
print(original_df)
print()
```

```
# Add some missing and duplicate values
df = df.append({'Name': 'Alice', 'Salary': 60000, 'Department': 'Marketing'}, ignore_index=True)
df = df.append({'Name': 'Karen', 'Salary': None, 'Department': 'HR'}, ignore_index=True)

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

# Print the modified data
print("Modified data:")
print(df)
```

SLIP 13

Q.1) Write a PHP script to create a chess board using CSS on table cells. [15]

SOLUTION:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .chess-board{
            border:solid black 2px;
        }
        .chess-board td {
            width: 50px;
            height: 50px;
        }
        .chess-board td.black {
            background-color: black;
        }
        .chess-board td.white {
            background-color: white;
        }
    </style>
</head>
<body>
    <table class="chess-board">
        <?php
        for ($row = 1; $row <= 8; $row++) {
            echo "<tr>";
            for ($col = 1; $col <= 8; $col++) {
                $color = ($row + $col) % 2 == 0 ? "black" : "white";
                echo "<td class='$color'></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) [10]

SOLUTION:

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

# Read the iris.csv file into a pandas DataFrame
df = pd.read_csv('iris.csv')

# Extract the petal length and petal width columns
petal_length = df['PetalLengthCm']
petal_width = df['PetalWidthCm']

# Create a scatter plot of the petal length and petal width
plt.scatter(petal_length, petal_width)
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
plt.title('Petal Length vs. Petal Width')

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

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

SOLUTION:

```
import numpy as np


# Define the array

array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

print("Original array = ",array)

# Flatten the array
```

```
flattened_array = array.flatten()

# Find the maximum and minimum value
max_value = flattened_array.max()
min_value = flattened_array.min()

# Print the results
print(f"The maximum value is {max_value}")
print(f"The minimum value is {min_value}")
```

SLIP 14

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

SOLUTION:

```
<!DOCTYPE html>
<html>
<head>
        <!-- Include Bootstrap CSS -->
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.c
ss"
        integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7S
k" crossorigin="anonymous">
   <style>

   .container{
      background-color: grey;
   }
   .col-4{
      border: solid 2px black;
      text-align: center;
   }
   </style>
</head>
<body>
        <!-- Your HTML content goes here -->
```

```
<div class="container">
  <div class="row">
    <div class="col-4">Column 1</div>
    <div class="col-4">Column 2</div>
    <div class="col-4">Column 3</div>
  </div>
</div>

</body>
</html>
```

Q. 2 A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array. [10]

SOUTION:

```
import numpy as np

# Define the array
array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(f"Original array = {array}")
# Flatten the array
flattened_array = array.flatten()

# Define the weights
weights = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

# Compute the weighted average along the specified axis (in this case, axis=0)
weighted_average = np.average(flattened_array, axis=None, weights=weights)
```

```
# Print the result
print(f"The weighted average is {weighted_average}")
```

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

SOLUTION:

```
import pandas as pd

# Read the advertising.csv file into a pandas DataFrame
df = pd.read_csv('advertising.csv')

# View the statistical details of the data
print(df.describe())
```

SLIP 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.

Find the reverse of given string.

SOLUTION:

Do From ass 3 set Aq1

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.

[10]

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()
```

```
# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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

# Create the lists of subject names and marks
subject_names = ['Math', 'Physics', 'Chemistry', 'Biology']
marks = [80, 90, 75, 85]

# Create the pie chart
plt.pie(marks, labels=subject_names)
```

```
# Add a title and show the plot
plt.title("Subject Marks")
plt.show()
```

SLIP 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) [15]

SOLUTION:

```
<?php


if (isset($_POST['submit']))  {
  $marks = $_POST['marks'];
  $subjects = ['Subject 1', 'Subject 2', 'Subject 3', 'Subject 4', 'Subject 5'];
  $total = 0;
  $percentage = 0;
  $grade = '';
  for ($i = 0; $i < count($marks); $i++) {
    $total += $marks[$i];
  }
  $percentage = $total / 500 * 100;
  if ($percentage >= 90) {
    $grade = 'A+';
  } elseif ($percentage >= 80) {
    $grade = 'A';
  } elseif ($percentage >= 70) {
    $grade = 'B+';
  } elseif ($percentage >= 60) {
    $grade = 'B';
```

```
  } else {
    $grade = 'F';
  }
}

?>
<!DOCTYPE html>
<html>
<head>
  <title>Marks Sheet</title>
</head>
<body>
  <?php if (isset($_POST['submit'])) { ?>
  <table>
    <tr>
      <th>Serial Number</th>
      <th>Subject</th>
      <th>Marks</th>
    </tr>
    <?php for ($i = 0; $i < count($marks); $i++) { ?>
    <tr>
      <td><?php echo $i + 1; ?></td>
      <td><?php echo $subjects[$i]; ?></td>
      <td><?php echo $marks[$i]; ?></td>
    </tr>
    <?php } ?>
```

```
    <tr>
      <td colspan="2">Total</td>
      <td><?php echo $total; ?></td>
    </tr>
    <tr>
      <td colspan="2">Percentage</td>
      <td><?php echo $percentage; ?></td>
    </tr>
    <tr>
      <td colspan="2">Grade</td>
      <td><?php echo $grade; ?></td>
    </tr>
  </table>
  <?php } else { ?>
  <form action="slip16b.php" method="post">
    <table>
      <tr>
        <th>Serial Number</th>
        <th>Subject</th>
        <th>Marks</th>
      </tr>
      <?php for ($i = 0; $i < 5; $i++) { ?>
      <tr>
        <td><?php echo $i?></td>
<td><?php echo $subjects[$i]; ?></td>
<td><input type="text" name="marks[]"></td>
```

```
</tr>
<?php } ?>
<tr>
<td colspan="3"><input type="submit" name="submit" value="Submit"></td>
</tr>
  </table>
</form>
<?php } ?>
</body>
</html>
```

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.

[10]

SOLUTION:

```
import matplotlib.pyplot as plt

# Define the subject names and marks
subjects = ['Math', 'Physics', 'Chemistry', 'Biology', 'English']
marks = [80, 75, 70, 65, 60]

# Create a pie chart
plt.pie(marks, labels=subjects, autopct='%1.1f%%')
plt.title('Marks Obtained in Subjects')
plt.show()
```

```
# Create a new figure
plt.figure()

# Create a bar chart
plt.bar(subjects, marks)
plt.xlabel('Subjects')
plt.ylabel('Marks')
plt.title('Marks Obtained in Subjects')
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.

SOLUTION:

```
import pandas as pd

# create a dictionary with the students' information
students_dict = {
    'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve', 'Frank'],
    'graduation_percentage': [80, 75, 90, 70, 60, 85],
    'age': [22, 25, 21, 23, 24, 20]
}

# create a data frame from the dictionary
df = pd.DataFrame(students_dict)
print(f"\nThe data frame is\n")
print(df)
```

```
# calculate and display the average age of the students
average_age = df['age'].mean()
print(f'Average age: {average_age:.2f}')

# calculate and display the average graduation percentage
average_graduation_percentage = df['graduation_percentage'].mean()
print(f'Average graduation percentage: {average_graduation_percentage:.2f}')
```

SLIP 17

) Write a PHP script to sort the following associative array :
array(“Sagar"=>"31","Vicky"=>"41","Leena"=>"39","Ramesh"=>"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 [15]

SOLUTION:

```
<?php


$array = array("Sagar" => "31", "Vicky" => "41", "Leena" => "39", "Ramesh" =>
"40");


// Sort array in ascending order by value

asort($array);

echo "Ascending order by value:<br>";

print_r($array);


echo "<br><br>";


// Sort array in ascending order by key

ksort($array);

echo "Ascending order by key:<br>";

print_r($array);


echo "<br><br>";
```

```
// Sort array in descending order by value
arsort($array);
echo "Descending order by value:<br>";
print_r($array);

echo "<br><br>";

// Sort array in descending order by key
krsort($array);
echo "Descending order by key:<br>";
print_r($array);

?>
```

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

SOLUTION:

```
import matplotlib.pyplot as plt
from sklearn import datasets

# Load the iris dataset
iris = datasets.load_iris()

# Extract the two features that you want to compare
feature1 = iris.data[:, 0]
feature2 = iris.data[:, 1]
```

```
# Draw a scatter plot comparing the two features
plt.scatter(feature1, feature2, c=iris.target, cmap='viridis')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Scatter plot of Feature 1 vs Feature 2')
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.

SOLUTION:

```
import pandas as pd

# Create a dictionary with the data for the data frame
data = {'name': ['John', 'Jane', 'Bob', 'Alice', 'Mike', 'Samantha', 'Charles', 'Emma', 'Edward', 'Emily'],
        'age': [32, 34, 28, 41, 37, 29, 45, 40, 35, 37],
        'salary': [50000, 60000, 45000, 80000, 55000, 51000, 90000, 72000, 62000, 74000],
        'department': ['Marketing', 'HR', 'IT', 'Sales', 'Marketing', 'HR', 'IT', 'Sales', 'Marketing', 'HR']}

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

# View the data frame
print(df)
```

SLIP 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. [15]

SOLUTION:

```
<?php


// Declare an associative array
$arr = array(
  "key1" => "value1",
  "key2" => "value2",
  "key3" => "value3"
);


do {
  // Display menu
  echo "Menu:\n";
  echo "1. Reverse the order of each element's key-value pair\n";
  echo "2. Traverse the element in an array in random order\n";
  echo "3. Convert the array elements into individual variables\n";
  echo "4. Display the elements of an array along with key\n";
  echo "Enter your choice (0 to exit): ";
```

```
// Read user input
$choice = trim(fgets(STDIN));

// Perform the requested operation
switch ($choice) {
  case 1:
    // Reverse the order of each element's key-value pair
    $reversed_arr = array_flip($arr);
    print_r($reversed_arr);
    break;
  case 2:
    // Traverse the element in an array in random order
    shuffle($arr);
    print_r($arr);
    break;
  case 3:
    foreach ($arr as $key => $value) {
      ${$key} = $value;
    }
    echo "\$key1 = $key1\n";
    echo "\$key2 = $key2\n";
    echo "\$key3 = $key3\n";
    break;
  //   break;
  case 4:
    // Display the elements of an array along with key
```

```
    foreach ($arr as $key => $value) {
      echo "$key: $value\n";
    }
    break;
  case 0:
    // Exit the program
    break;
  default:
    echo "Invalid choice\n";
 }
} while ($choice != 0);

?>
```

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) [10]

SOLUTION:

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

# Load the iris dataset into a Pandas data frame
df = pd.read_csv('iris.csv')

# Extract the features and target variables from the data frame
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
```

```
# Create a box plot for each feature, colored by the target variable
# Create a box plot for each feature, colored by the target variable
fig, axs = plt.subplots(1, 4, figsize=(12, 3))
colors = {'Iris-setosa': 'red', 'Iris-versicolor': 'green', 'Iris-virginica': 'blue'}

for i, (feature, color) in enumerate(zip(X.columns, colors.values())):
    axs[i].boxplot(X[feature], labels=[feature], notch=True, patch_artist=True,
                   boxprops=dict(facecolor=color, color=color))

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

SOLUTION:

```
import pandas as pd

# Load the dataset into a dataframe
df = pd.read_csv('iris.csv')

# Print the first 5 rows
print(df.head(5))

# Print the last 5 rows
print(df.tail(5))
# Print 10 random rows
print(df.sample(10))
```

SLIP 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. [15]

SOLUTION:

```
<?php

// Accept the two strings from the user
$sentence = readline("Enter a sentence: ");
$word = readline("Enter a word: ");

// Delete a small part from the first string
echo "Enter the position to start deletion: ";
$pos = readline();
$pos = intval($pos); // convert string to integer

echo "Enter the number of characters to delete: ";
$numChars = readline();
$numChars = intval($numChars); // convert string to integer

$sentence = substr_replace($sentence, "", $pos, $numChars);
echo "After deletion: $sentence\n";
```

```
// Insert the given small string in the given big string at a specified position
echo "Enter the position to insert the word: ";
$pos = readline();
$pos = intval($pos); // convert string to integer

$sentence = substr_replace($sentence, $word, $pos, 0);
echo "After insertion: $sentence\n";

// Replace some characters/words from the given big string with the given small string at a specified position
echo "Enter the position to start replacement: ";
$pos = readline();
$pos = intval($pos); // convert string to integer

echo "Enter the number of characters to replace: ";
$numChars = readline();
$numChars = intval($numChars); // convert string to integer

$sentence = substr_replace($sentence, $word, $pos, $numChars);
echo "After replacement: $sentence\n";

?>
```

Q.2) Write a Python program

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.

SOLUTION:

```
import pandas as pd


# Create a dictionary with the data for the dataframe
data = {'name': ['John', 'Jane', 'Bob', 'Alice', 'Mike', 'Samantha', 'Charles', 'Emma', 'Edward', 'Emily'],
        'age': [32, 34, 28, 41, 37, 29, 45, 40, 35, 37],
        'percentage': [70, 80, 65, 90, 75, 71, 95, 82, 77, 84]}


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


# View the dataframe
print(df)
# Print the shape of the dataframe
print(df.shape)


# Print the number of rows and columns
print(f'Number of rows: {df.shape[0]}')
print(f'Number of columns: {df.shape[1]}')
```

```
# Print the data types of the columns
print(df.dtypes)

# Print the feature names
print(df.columns)

# Print the description of the data
print(df.describe())
# Add 5 rows with duplicate values and missing values
df2 = pd.DataFrame({'name': ['John', 'Jane', 'Bob', 'Alice', 'Mike'],
                    'age': [32, 34, 28, 41, 37],
                    'percentage': [70, 80, 65, 90, 75]})
df = df.append(df2, ignore_index=True)

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

# Display the data
print(df)
```

SLIP 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. [15]

SOLUTION:

```
<?php


// define an associative array
$array = array(
  'a' => 1,
  'b' => 6,
  'c' => 2,
  'd' => 0,
  'e' => 10,
  'f' => 3,
  'g' => 4
);


echo "The OG array = ";
print_r($array);
// display menu
echo "Menu: \n";
echo "1. Split array into chunks \n";
echo "2. Sort array by values \n";
```

```
echo "3. Filter even elements from array \n";
echo "Enter your choice: ";

// read user input
$choice = trim(fgets(STDIN));

// perform the selected operation
switch ($choice) {
  case 1:
    // split array into chunks
    $chunk_size = 2; // specify chunk size
    $chunks = array_chunk($array, $chunk_size, true); // split array
    print_r($chunks); // print chunks
    break;
  case 2:
    // sort array by values
    asort($array); // sort array
    print_r($array); // print sorted array
    break;
  case 3:
    // filter even elements from array
    $even_elements = array_filter($array, function($value) {
      return $value % 2 == 0;
    });
    print_r($even_elements); // print even elements
    break;
```

```
  default:
    echo "Invalid choice. \n";
}

?>
```

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. [10]

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
```

```
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
plt.show()
```

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

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

# Generate a random array of 50 integers
data = np.random.randint(low=0, high=100, size=50)

# Line chart
plt.plot(data, color='blue', linewidth=2)
plt.title('Line chart')
plt.xlabel('Index')
plt.ylabel('Value')
plt.show()

# Scatter plot
plt.scatter(range(len(data)), data, color='green')
plt.title('Scatter plot')
```

```
plt.xlabel('Index')
plt.ylabel('Value')
plt.show()

# Histogram
plt.hist(data, bins=10, color='red')
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()

# Box plot
plt.boxplot(data, notch=True, patch_artist=True,
boxprops=dict(facecolor='purple', color='purple'))
plt.title('Box plot')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
# Add two outliers to the data
data = np.append(data, [200, 300])

# Box plot with outliers
plt.boxplot(data, notch=True, patch_artist=True,
boxprops=dict(facecolor='purple', color='purple'))
plt.title('Box plot with outliers')
plt.xlabel('Value')
plt.ylabel('Frequency')
```

```
plt.show()
```

SLIP 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.

SOLUTION:

```
<?php
$month_temp = "78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 81, 76, 73,
68, 72, 73, 75, 65, 74, 63, 67, 65, 64, 68, 73, 75, 79, 73";
echo "15 Temperature = ";
print_r($month_temp);
$temp_array = explode(',', $month_temp);
$tot_temp = 0;
$temp_array_length = count($temp_array);
foreach($temp_array as $temp)
{
 $tot_temp += $temp;
}
 $avg_high_temp = $tot_temp/$temp_array_length;
 echo "Average Temperature is : ".$avg_high_temp." ";
sort($temp_array);
echo " List of five lowest temperatures :";
for ($i=0; $i< 5; $i++)
{
echo $temp_array[$i].", ";
}
echo "List of five highest temperatures :";
```

```
for ($i=($temp_array_length-5); $i< ($temp_array_length); $i++)
{
echo $temp_array[$i].", ";
}
?>
```

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. [10]

SOLUTION:

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

# Load the Iris dataset into a Pandas dataframe
df = pd.read_csv('iris.csv')

# Extract the target variable (the species of Iris) from the dataframe
y = df['Species']

# Get the frequency of each species
species_counts = y.value_counts()

# Create a bar plot
plt.bar(species_counts.index, species_counts.values, color='purple')
plt.title('Frequency of Iris Species')
plt.xlabel('Species')
plt.ylabel('Frequency')
plt.show()
```

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

SOLUTION:

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

# Load the Iris dataset into a Pandas dataframe
df = pd.read_csv('iris.csv')

# Extract the features and target variables from the dataframe
X = df.iloc[:, :-1]
Y = df.iloc[:, -1]

# Create a histogram for each feature, colored by the target variable
colors = {'Iris-setosa': 'red', 'Iris-versicolor': 'green', 'Iris-virginica': 'blue'}

for i, feature in enumerate(X.columns):
    plt.hist(X[feature], bins=10, color=colors[Y[i]])
    plt.title(f'Histogram of {feature}')
    plt.xlabel(feature)
    plt.show()
```

SLIP 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 [15]

SOLUTION:

```
<html>
<body bgcolor=skyblue>
<form action="slip23.php" method="post">
Enter choice :
<input type="radio" name="ch" value=4> Insert element in queue<br>
<input type="radio" name="ch" value=5> Delete element from queue <br>
<input type="radio" name="ch" value=6> Display content of queue <br>
<br>

<input type="submit" value="submit">
<input type="reset" value="reset">
</body>

</html>
<html>
<body bgcolor="gold">
<?php
$choice=$_POST['ch'];
{
```

```
    $arr=array(1,2,3,4,5,6,7,8,9,10);
    switch($choice)
    {
        case 1:
            array_unshift($arr,"10");
            print_r($arr);
            break;
        case 2:
            $ele=array_shift($arr);
            echo "Deleted element : $ele";
            break;
        case 3:
            print_r($arr);
            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.)

SOLUTION:

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

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

# Split the dataset into features and target
X = df.drop('quality', axis=1)
y = df['quality']

# Rescale the features using MinMaxScaler
scaler = MinMaxScaler()
X_rescaled = scaler.fit_transform(X)

# Standardize the features using StandardScaler
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X)

# Normalize the features using Normalizer
scaler = Normalizer()
X_normalized = scaler.fit_transform(X)
```

SLIP 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 [15]

SOLUTION:

```
<html>
<body bgcolor=skyblue>
<form action="slip23.php" method="post">
Enter choice :
<br><input type="radio" name="ch" value=1> Insert element in stack <br>
<input type="radio" name="ch" value=2> Delete element from stack <br>
<input type="radio" name="ch" value=3> Display content of stack <br>
<br>

<input type="submit" value="submit">
<input type="reset" value="reset">
</body>

</html>
<html>
<body bgcolor="gold">
<?php
$choice=$_POST['ch'];
```

```
{
    $arr=array(1,2,3,4,5,6,7,8,9,10);
    switch($choice)
    {
        case 1:
            array_push($arr,10);
            print_r($arr);
            break;
        case 2:
            $ele=array_pop($arr);
            echo "Poped element : $ele";
            break;
        case 3:
            print_r($arr);
            break;


    }
}
?>
```

Q.2) Dataset Name: winequality-red.csv

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)

SOLUTION:

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

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

# Split the dataset into features and target
X = df.drop('quality', axis=1)
y = df['quality']

# Rescale the features using MinMaxScaler
scaler = MinMaxScaler()
X_rescaled = scaler.fit_transform(X)

# Standardize the features using StandardScaler
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X)

# Binarize the features using Binarizer
scaler = Binarizer(threshold=0.5)
X_binarized = scaler.fit_transform(X)
```

SLIP 24

Q.1) Write a PHP program to read two file names from user and append content of first file into second file. [15]

SOLUTION:

```
<?php

// Prompt the user for the name of the first file
echo "Enter the name of the first file: ";
$file1 = readline();

// Prompt the user for the name of the second file
echo "Enter the name of the second file: ";
$file2 = readline();

// Open the first file for reading
$handle1 = fopen($file1, "r");

// Open the second file for appending
$handle2 = fopen($file2, "a");

// Read the contents of the first file
$contents1 = fread($handle1, filesize($file1));

// Append the contents of the first file to the second file
fwrite($handle2, $contents1);
```

```
// Close the handles
fclose($handle1);
fclose($handle2);

echo "Content of first file appended to second file successfully!";

?>
```

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. [10]

SOLUTION:

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

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

# Count the number of occurrences of each species
species_counts = df['Species'].value_counts()

# Get the species names
species_names = species_counts.index

# Get the frequency of each species
species_freq = species_counts.values

# Create the bar plot
```

```
plt.bar(species_names, species_freq)
plt.xlabel('Species')
plt.ylabel('Frequency')
plt.title('Iris Species Frequency')
plt.show()
```

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

SOLUTION:

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

# Load the Iris dataset into a Pandas dataframe
df = pd.read_csv('iris.csv')

# Extract the features and target variables from the dataframe
X = df.iloc[:, :-1]
Y = df.iloc[:, -1]

# Create a histogram for each feature, colored by the target variable
colors = {'Iris-setosa': 'red', 'Iris-versicolor': 'green', 'Iris-virginica': 'blue'}

for i, feature in enumerate(X.columns):
    plt.hist(X[feature], bins=10, color=colors[Y[i]])
    plt.title(f'Histogram of {feature}')
    plt.xlabel(feature)
    plt.show()
```

SLIP 25

Q.1) Write a menu driven program to perform various file operations. Accept filename from user. [15]

a) Display type of file.

b) Display last modification time of file

c) Display the size of file

d) Delete the file

SOLUTION:

```
<html>


<form action="slip25.php" method="post">


Enter 1st file name<input type="text" name="file" value=""><br>


Enter 2nd file name<input type="text" name="file1" value=""><br>


Enter directory name<input type="text" name="dir" value=""><br>


<input type="radio" name="b" value="1">1.DISPLAY SIZE OF FILE<br>


<input type="radio" name="b" value="2">2.DISPLAY LAST
ACCESSED,CHANGED,MODIFIED TIME OF FILE<br>


<input type="radio" name="b" value="3">3.DISPLAY DETAILS ABOUT OWNER
AND USER OF FILE<br>
```

```
<input type="radio" name="b" value="4">4.DISPLAY TYPE OF FILE<br>

<input type="radio" name="b" value="5">5.COPY A FILE<br>

<input type="radio" name="b" value="6">6.TRAVERSE A DIRECTORY<br>

<input type="submit" name="submit" value="submit"><br>

</form>

</html>
<?php

$file=$_POST['file'];

$file1=$_POST['file1'];

$dir=$_POST['dir'];

$c=$_POST['b'];

$db=opendir($dir);

switch($c)
```

```
{

case 1 :

        $size=filesize("$file");

        echo "the size of file is:$size";

        break;

case 2:

    $a=fileatime($file);

    echo "last accessed time of file  is :$a<br>";

    $c=filectime($file);

    echo "last changed time of file $c";

    $m=filemtime($file);

    echo "last modified time of file $m";

    break;
```

```
case 3:

    $o=fileowner($file);

    echo "owner  of file is $o";

    break;

case 4:

    $t=filetype($file);

    echo "type of file is :$t";

        break;

    case 5:

        copy("$file","$file1");

        echo"copy successfully";

        break;

    case 6:
```

```
        while($a=readdir($db))

        {

          echo"$a<br>";

        }

        break;

    default:"invalid choice";

}

?>
```

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. [10]

SOLUTION:

```
import random

import matplotlib.pyplot as plt


# Generate a random array of 50 integers

data = [random.randint(0, 100) for _ in range(50)]
```

```
# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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.

SOLUTION:

```
import matplotlib.pyplot as plt

# Create the lists of subject names and marks
subject_names = ['Math', 'Physics', 'Chemistry', 'Biology']
marks = [80, 90, 75, 85]
```

```
# Create the pie chart
plt.pie(marks, labels=subject_names)

# Add a title and show the plot
plt.title("Subject Marks")
plt.show()
```

SLIP 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.

SOLUTION:

```
<html>
<head>
<style>
table {
  border-collapse: collapse;
}

table, th, td {
  border: 1px solid black;
  padding: 5px;
}

</style>
</head>
<body>
<div style="text-align: center;">
```

```
  <form method="GET" action="slip26.php">
    Hospital Name: <input type="text" name="hname">
    <input type="submit" value="Submit">
  </form>
</div>

<?php
$hname = $_GET['hname'];

$db = pg_connect("host=localhost port=5432 dbname=Slips user=hp
password=");
if ($db) {
  $query = "SELECT * FROM doc WHERE dno IN (SELECT dno FROM doc_hosp
WHERE hno IN (SELECT hno FROM hosp WHERE hname='$hname'))";
  $resultSet = pg_query($db, $query);
  echo "<h1>Doctors from hospital $hname are:</h1>";
  echo "<table style='height: 400; width: 400;'>";
  echo "<tr><th>Dno</th>";
  echo "<th>Name</th>";
  echo "<th>Address</th>";
  echo "<th>City</th></tr>";
  if (pg_num_rows($resultSet) > 0) {
    while ($row = pg_fetch_array($resultSet)) {
      echo "<tr><td>" . $row['dno'] . "</td>";
      echo "<td>" . $row['dname'] . "</td>";
      echo "<td>" . $row['addr'] . "</td>";
      echo "<td>" . $row['city'] . "</td></tr>";
```

```
    }
  } else {
    echo "<tr><td colspan='4'>No doctors found</td></tr>";
  }
  echo "</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. [10]

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()

# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
```

```
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
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.

SOLUTION:

```
import matplotlib.pyplot as plt

# Create the lists of subject names and marks
subject_names = ['Math', 'Physics', 'Chemistry', 'Biology']
marks = [80, 90, 75, 85]

# Create the pie chart
plt.pie(marks, labels=subject_names)

# Add a title and show the plot
plt.title("Subject Marks")
plt.show()
```

SLIP 27

Q.1) Write a PHP program to read two file names from user and copy the content of first file into second file. [15]

SOLUTION:

```
<?php

// Prompt the user for the name of the first file
echo "Enter the name of the first file: ";
$file1 = readline();

// Prompt the user for the name of the second file
echo "Enter the name of the second file: ";
$file2 = readline();

// Open the first file for reading
$handle1 = fopen($file1, "r");

// Open the second file for appending
$handle2 = fopen($file2, "a");

// Read the contents of the first file
$contents1 = fread($handle1, filesize($file1));

// Append the contents of the first file to the second file
fwrite($handle2, $contents1);
```

```
// Close the handles
fclose($handle1);
fclose($handle2);

echo "Content of first file appended to second file successfully!";

?>
```

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

SOLUTION:

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

# Load the Data dataset
data = pd.read_csv("Data.csv")

# OneHot encode the Country column
one_hot = pd.get_dummies(data['Country'])

# Label encode the Purchased column
le = LabelEncoder()
```

```
le.fit(data['Purchased'])
label_encoded = le.transform(data['Purchased'])

# Add the encoded columns to the original dataset
data = pd.concat([data, one_hot, pd.DataFrame(label_encoded, columns=['Purchased'])], axis=1)

# Drop the original Country and Purchased columns
data.drop(['Country', 'Purchased'], axis=1, inplace=True)
```

SLIP 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) [15]

SOLUTION:

```
<?php

// Open the student.dat file in read mode
$file = fopen("student.dat", "r");

// Read the data from the file line by line
while (($line = fgets($file)) !== false) {
  // Split the line into an array of fields
  $fields = explode(",", $line);

  // Assign the fields to variables
  $rollno = $fields[0];
  $name = $fields[1];
  $os = $fields[2];
  $wt = $fields[3];
  $ds = $fields[4];
  $python = $fields[5];
  $java = $fields[6];
  $cn = $fields[7];

  // Calculate the total marks and the percentage
```

```
$total = $os + $wt + $ds + $python + $java + $cn;
$percentage = ($total / 600) * 100;

// Display the data in a tabular format
echo "<table>";
echo "<tr>";
echo "<td>Roll No:</td>";
echo "<td>$rollno</td>";
echo "</tr>";
echo "<tr>";
echo "<td>Name:</td>";
echo "<td>$name</td>";
echo "</tr>";
echo "<tr>";
echo "<td>OS:</td>";
echo "<td>$os</td>";
echo "</tr>";
echo "<tr>";
echo "<td>WT:</td>";
echo "<td>$wt</td>";
echo "</tr>";
echo "<tr>";
echo "<td>DS:</td>";
echo "<td>$ds</td>";
echo "</tr>";
echo "<tr>";
```

```
  echo "<td>Python:</td>";
  echo "<td>$python</td>";
  echo "</tr>";
  echo "<tr>";
  echo "<td>Java:</td>";
  echo "<td>$java</td>";
  echo "</tr>";
  echo "<tr>";
  echo "<td>CN:</td>";
  echo "<td>$cn</td>";
  echo "</tr>";
  echo "<tr>";
  echo "<td>Total:</td>";
  echo "<td>$total</td>";
  echo "</tr>";
  echo "<tr>";
  echo "<td>Percentage:</td>";
  echo "<td>$percentage</td>";
  echo "</tr>";
  echo "</table>";
}

// Close the file
fclose($file);

?>
```

STUDENT.DAT file format

01,nik,78,90,85,90,88,90

01,nik,78,90,85,90,88,90

01,nik,78,90,85,90,88,90

01,nik,78,90,85,90,88,90

01,nik,78,90,85,90,88,90

01,nik,78,90,85,90,88,90

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.

SOLUTION:

```
import pandas as pd


# Create a dictionary with the data for the dataframe
data = {'name': ['John', 'Jane', 'Bob', 'Alice', 'Mike', 'Samantha', 'Charles', 'Emma', 'Edward', 'Emily'],
        'age': [32, 34, 28, 41, 37, 29, 45, 40, 35, 37],
        'percentage': [70, 80, 65, 90, 75, 71, 95, 82, 77, 84]}


# Create the dataframe
```

```
df = pd.DataFrame(data)

# View the dataframe
print(df)
# Print the shape of the dataframe
print(df.shape)

# Print the number of rows and columns
print(f'Number of rows: {df.shape[0]}')
print(f'Number of columns: {df.shape[1]}')

# Print the data types of the columns
print(df.dtypes)

# Print the feature names
print(df.columns)

# Print the description of the data
print(df.describe())
# Add 5 rows with duplicate values and missing values
df2 = pd.DataFrame({'name': ['John', 'Jane', 'Bob', 'Alice', 'Mike'],
                    'age': [32, 34, 28, 41, 37],
                    'percentage': [70, 80, 65, 90, 75]})
df = df.append(df2, ignore_index=True)

# Add a column 'remarks' with empty values
```

```
df['remarks'] = ''
# Display the data
print(df)
```

SLIP 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.

SOLUTION:

```
<html>
<body>
<form action="a5a1.php" method="POST">
Enter Department name : <input type="text" name="dept_name" /></br>
<input type="submit"/>
</form>
</body>
</html>
<?php

$dept_name = $_POST['dept_name'];

$con = pg_connect( "dbname=test" );
if( !$con ) {
echo 'Connection not established';
```

```
exit;
}

$dept_insert = "insert into dept values(1,'HR','1st Floor') ,(2, 'Comp sci', '2nd floor'),(3,'Maths','2nd
Floor'),(4,'Electronics','3rd Floor')";

$res = pg_query($con,$dept_insert);

$emp_insert = "insert into employee values( 1, 'abc','Pune','7894356','20000',1),( 2, 'abc','Pune','7894356','10000', 1),( 3,
'abc','Pune','7894356','30000',3),( 4, 'abc','Pune','7894356','120000',2)";

$res1 = pg_query($con,$emp_insert);

$strQuery = "select max(salary), min(salary), sum(salary) from employee where d_no in(select d_no from dept where
dname = '$dept_name')";

$result = pg_query($con,$strQuery);

$row = pg_fetch_row($result);

?>
<html>
<body>
<table>
```

```
<tr>
<th>Maximum salary</th>
<th>Minimum salary</th>
<th>Sum</th>
</tr>
<tr>
<td><?php echo $row[0];?></td>
<td><?php echo $row[1];?></td>
<td><?php echo $row[2];?></td>
</tr>
</table>
</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

SOLUTION:

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

# Load the Data dataset
data = pd.read_csv("Data.csv")

# OneHot encode the Country column
one_hot = pd.get_dummies(data['Country'])
```

```
# Label encode the Purchased column
le = LabelEncoder()
le.fit(data['Purchased'])
label_encoded = le.transform(data['Purchased'])

# Add the encoded columns to the original dataset
data = pd.concat([data, one_hot, pd.DataFrame(label_encoded,
columns=['Purchased'])], axis=1)

# Drop the original Country and Purchased columns
data.drop(['Country', 'Purchased'], axis=1, inplace=True)
```

SLIP 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.

SOLUTION:

```
html>
<body>
<form method="get" action="studcomp.php">
<fieldset>
        <legend>Enter Competition Name :</legend><br>
<input type="text" name="cnm"><br><br>
</fieldset>
<div align="center">
<input type="submit" value="Show Result">
</div>
</form>
</body>
</html>
<!--StudCom.php-->
```

```
<?php
	$cnames=$_GET['cnm'];

		$hn="localhost";
		$un="root";
		$pass="";
		$db="students";

		$link=mysqli_connect($hn,$un,$pass,$db);
	if(!$link)
	{
		die('Connection Failed:'.mysqli_error());
	}
	//$sql="SELECT * FROM student WHERE CNo IN (SELECT CNo FROM competition WHERE CName = '".$cname."')";
	$sql="select * from student,competition,studcomp where student.Sid=studcomp.Sid and competition.CNo=studcomp.CNo and rank='1' and CName='".$cnames."'";
	$res=mysqli_query($link,$sql);

	if(mysqli_num_rows($res)>0)
	{
		while($row=mysqli_fetch_assoc($res))
	{
			echo"Stud No : ".$row['Sid']."<br>"."  Name  : ".$row['SName']."<br>";
			echo"Class :  ".$row['SClass']."<br>";
```

```
                    echo"-------------------------"."<br>";
                }
        }
        else
        {
                echo"error";
        }
        mysqli_close($link);

?>
```

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 appropriate color, labels and styling options.

SOLUTION:

```
import random
import matplotlib.pyplot as plt

# Generate a random array of 50 integers
data = [random.randint(0, 100) for _ in range(50)]

# Line chart
plt.plot(data, '-', color='red', label='Line chart')
plt.legend()
plt.show()
```

```
# Scatter plot
plt.scatter(range(50), data, color='green', label='Scatter plot')
plt.legend()
plt.show()

# Histogram
plt.hist(data, bins=10, color='blue', label='Histogram')
plt.legend()
plt.show()

# Box plot
plt.boxplot(data, patch_artist=True, labels=['Box plot'])
plt.show()
```

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

SOLUTION:

```
import matplotlib.pyplot as plt

# Create the lists of subject names and marks
subject_names = ['Math', 'Physics', 'Chemistry', 'Biology']
marks = [80, 90, 75, 85]

# Create the pie chart
plt.pie(marks, labels=subject_names)
```

```
# Add a title and show the plot
plt.title("Subject Marks")
plt.show()
```