

# DELHI TECHNOLOGICAL UNIVERSITY

## DEPARTMENT OF APPLIED MATHEMATICS



---

### WEB TECHNOLOGY (MC-320)

### LAB FILE

---

#### Submitted to:

Dr. Dinesh Udar  
Assistant Professor  
Department of Mathematics  
Delhi Technological University

#### Submitted by:

Kunal Sinha  
2K17/CO/164  
Computer Science(A3)

# VISION

---

To emerge as a centre of excellence and eminence by imparting futuristic technical education with solid mathematical background in keeping with global standards, making our students technologically and mathematically competent and ethically strong so that they can readily contribute to the rapid advancement of society and mankind.

# MISSION

---

1. To achieve academic excellence through innovative teaching and learning practices.
2. To improve the research competence to address social needs.
3. To inculcate a culture that supports and reinforces ethical, professional behaviors for a harmonious and prosperous society.
4. Strive to make students understand, appreciate and gain mathematical skills and develop logic, so that they are able to contribute intelligently in decision making which characterizes our scientific and technological age.

# **Programme Educational Objectives (PEOs)**

---

1. To prepare graduates with a solid foundation in Engineering, Mathematical Science and technology for a successful career in Mathematics & Computing/Finance/Computer Engineering fields.
2. To prepare graduates to become effective collaborators/innovators, who could ably address tomorrow's social, technical and engineering challenges.
3. To enrich graduates with integrity and ethical values so that they become responsible engineers.

## **Programme Specific Outcomes (PSOs)**

---

1. Design and analyze the mathematical models for the problem related to industry and socio-economic world.
2. Develop an algorithm to perform tasks related to research/training for the industry and education.
3. Develop aptitude for managerial capacity and research & development

# INDEX

S.No	Experiment	Date	Signature	Remarks
1	Design a webpage with HTML, document features and tag attributes having facilities for inserting table, form and text boxes.	10-01-2020		
2	Design a webpage for inserting image , voice and video clips.	17-01-2020		
3	Getting files from servers via FTP request, response,entity headers	24-01-2020		
4	Getting files from server via HTTP using request , response,general and entity headers	24-01-2020		
5	Posting messages to server via electronic mail	07-02-2020		
6	Posting messages to server via SMTP	21-02-2020		
7	Design a webpage for enrollment of students which accepts complete details of students.	10-04-2020		
8	To implement CGI design for transacting with database from client system	17-04-2020		

## EXPERIMENT 1

- **AIM :** Design a webpage with HTML, document features and tag attributes having facilities for inserting table, form and text boxes.
- **DESCRIPTION :** HTML is the standard markup language for Web pages. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

The <!DOCTYPE html> declaration defines this document to be HTML5

The <html> element is the root element of an HTML page.

The <head> element contains meta information about the document.

The <title> element specifies a title for the document.

The <body> element contains the visible page content.

The <h1> element defines a large heading.

The <p> element defines a paragraph.

- **CODE**

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Document</title>
```

```
</head>
```

```
<body>
```

```
  <h2>Example Form</h2>
```

```
  <form>
```

```
    <p>
```

```
      <label>Name: </label><input type="text" name="value" placeholder="Enter Name">
```

```
      <label>Date of Birth: </label> <input type="date" name="ip" value="0">
```

```
    </p>
```

```
    <p>
```

```
      <label>E-Mail: </label><input type="email" name="e-mail-address"
placeholder="example@gmail.com">
```

```
    </p>
```

```
    <p>
```

```
      <label>Password:</label><input type="password">
```

```
      <label>Continent: </label>
```

```
        <select name="Country">
```

```
          <option value="India">India</option>
```

```
          <option value="India">France</option>
```

```
        <option value="Sweden">Sweden</option>
    </select>
</p>
<p>Some lines about you</p>
    <textarea id="info" placeholder="Description" rows="5" cols="20"></textarea><br>
<label>Table </label>
<table border=1px solid black>
    <tr>
        <th>First Name</th>    <th>Last Name</th>    <th>Age</th>
    </tr>
    <tr>
        <td>Jill</td>        <td>Smith</td>        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>        <td>Jackson</td>        <td>94</td>
    </tr>
    <tr>
        <td>John</td>    <td>Doe</td>        <td>80</td>
    </tr>
</table>
<br>
    <input type="submit" name='sub' onclick="<a href='www.djangoproject.com'></a>">
    <input type = 'reset'>
</form>
</body>
</html>
```

- **OUTPUT**

**Example Form**

Name:  Date of Birth:

E-Mail:

Password:

Continent:

Some lines about you

Table

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

- **FINDINGS AND LEARNINGS :** From this experiment we learnt what is HTML and how to use the different types of HTML tags, their functions and how they are used to produce web pages. We also made a small form using different HTML tags and attributes.

## EXPERIMENT 2

- **AIM :** Design a webpage for inserting image , voice and video clips
- **DESCRIPTION :** The HTML <video> element specifies a standard way to embed a video in a web page. The HTML <audio> element specifies a standard way to embed audio in a web page. The controls attribute adds video and audio controls, like play, pause, and volume. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video and audio files which the browser may choose from.

The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

In HTML, images are defined with the <img> tag. The <img> tag is empty, it contains attributes only, and does not have a closing tag. The src attribute specifies the URL of the image.

- **CODE:**

```
<html>
<head>
  <title>Images , Video and Audio</title>
</head>
<body>
<h2>HTML Image</h2>
  

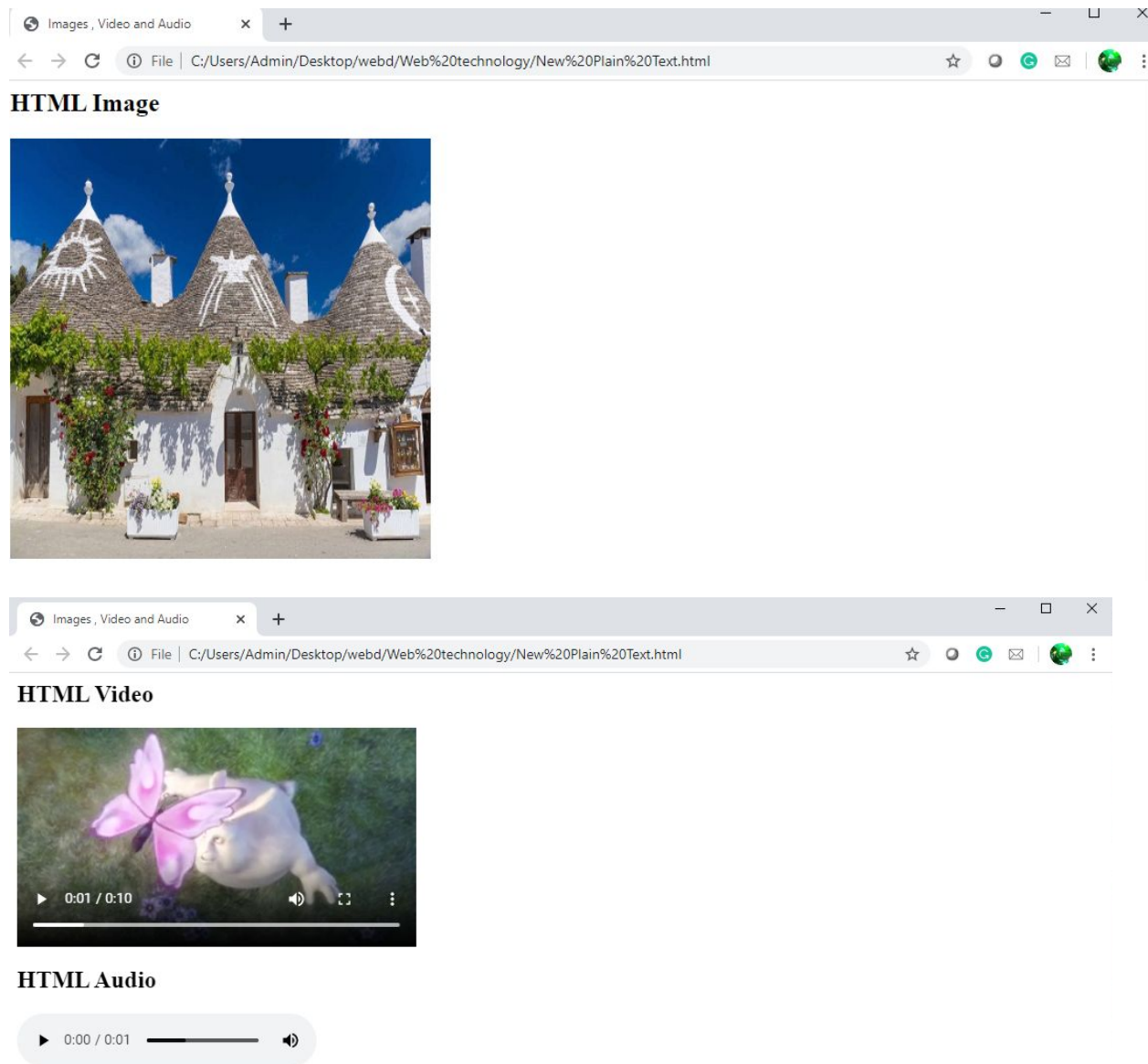
<h2>HTML Video</h2>
  <video width="400" controls>
    <source src="mov_bbb.mp4" type="video/mp4">
    Your browser does not support HTML5 video.
  </video>

<h2>HTML Video</h2>
  <audio controls>
    <source src="horse.ogv" type="audio/ogg">
    Your browser does not support the audio element.
  </audio>

</body>
</html>
```



- **OUTPUT**



- **FINDINGS AND LEARNINGS :** From this experiment we learnt about the image, audio and video tags and how they are used in web pages with other control features like play and pause. A text can also be written which gets displayed if the browser does not support the audio or video.

## EXPERIMENT 3

- **AIM**

Getting files from servers via FTP request, response, entity headers

- **DESCRIPTION :** FTP stands for file transfer protocol. It is a means of transferring files between computers of different types across a network. A protocol is a language that enables computers to speak to one another. FTP is used to make files and folders publicly available for transfer over the Internet. The general format for ftp is **ftp://name\_of\_ftp\_site/directory\_name/file\_name**.

This module defines the class FTP and a few related items. We used this to write Python programs that perform a variety of automated FTP jobs, such as mirroring other FTP servers. It is also used by the module urllib.request to handle URLs that use FTP.

- **CODE**

```
from ftplib import FTP
import ftplib
import os
"""A = FTP('ftp.debian.org')
A.login()
A.cwd('debian')
A.retrlines('LIST')"""
with FTP('speedtest.tele2.net') as ftp:
    try:
        ftp.login()
        print(ftp.getwelcome())
        A = []
        ftp.dir(A.append)
        #print(A)
        for x in A:
            print(x)
    except ftplib.all_errors as e:
        print('FTP error',e)
```

- **OUTPUT**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\ARNAV DTU\Semester 6\Web Technology> python -u "e:\ARNAV DTU\Semester 6\Web Technology\exp5.py"
220 (vsFTPD 3.0.3)
-rw-r--r-- 1 0 0 1073741824000 Feb 19 2016 100GB.zip
-rw-r--r-- 1 0 0 107374182400 Feb 19 2016 100GB.zip
-rw-r--r-- 1 0 0 102400 Feb 19 2016 100KB.zip
-rw-r--r-- 1 0 0 104857600 Feb 19 2016 100MB.zip
-rw-r--r-- 1 0 0 10737418240 Feb 19 2016 10GB.zip
-rw-r--r-- 1 0 0 10485760 Feb 19 2016 10MB.zip
-rw-r--r-- 1 0 0 1073741824 Feb 19 2016 1GB.zip
-rw-r--r-- 1 0 0 1024 Feb 19 2016 1KB.zip
-rw-r--r-- 1 0 0 1048576 Feb 19 2016 1MB.zip
-rw-r--r-- 1 0 0 209715200 Feb 19 2016 200MB.zip
-rw-r--r-- 1 0 0 20971520 Feb 19 2016 20MB.zip
-rw-r--r-- 1 0 0 2097152 Feb 19 2016 2MB.zip
-rw-r--r-- 1 0 0 3145728 Feb 19 2016 3MB.zip
-rw-r--r-- 1 0 0 524288000 Feb 19 2016 500MB.zip
-rw-r--r-- 1 0 0 52428800 Feb 19 2016 50MB.zip
-rw-r--r-- 1 0 0 524288 Feb 19 2016 512KB.zip
-rw-r--r-- 1 0 0 5242880 Feb 19 2016 5MB.zip
-rw-r--r-- 1 0 0 3145728 Feb 19 2016 3MB.zip
-rw-r--r-- 1 0 0 524288000 Feb 19 2016 500MB.zip
-rw-r--r-- 1 0 0 52428800 Feb 19 2016 50MB.zip
-rw-r--r-- 1 0 0 524288 Feb 19 2016 512KB.zip
-rw-r--r-- 1 0 0 5242880 Feb 19 2016 5MB.zip
drwxr-xr-x 2 105 108 561152 Apr 19 13:56 upload
PS E:\ARNAV DTU\Semester 6\Web Technology>
```

- **FINDINGS AND LEARNINGS :** From this experiment we learnt about file transfer protocol , its features and different applications. The file transfer protocol can access a wide range of information that is available online in forms of archives to be accessed by the common public. FTP servers provide a storage place for useful files and programs. It is command driven, therefore, commands like open, get, etc., are used.

## EXPERIMENT 4

- **AIM**

Getting files from server via HTTP using request , response,general and entity headers

- **DESCRIPTION :** Here we'll use pure JavaScript for getting files from the server via the request,response and entity headers AJAX: Used for modern web pages for transferring data without the requirement of reloading a web page. This AJAX Response is the object passed as the first argument of all Ajax requests callbacks. This is a wrapper around the native XMLHttpRequest object. It normalizes cross-browser issues while adding support for JSON via the responseJSON and headerJSON properties.

- **CODE :**

### HTML file

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>File transfer</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJ
wJ8ERdknLPMO" crossorigin="anonymous">
</head>
<body>
  <p id="head"></p>
  <div class="container">
    <p id='Fetch Paragraph'>
      Fetch content from here
    </p>
    <button id="FetchBtn" class="btn btn-primary" > Fetch</button>
    <ul id="list">
    </ul>
    <button id="SendBtn" class="btn btn-primary" >Send</button>
  </div>
</body>
<script src="exp4.js"></script>
</html>
```

### JavaScript file

```
console.log('exp4');
let FetchBtn = document.getElementById('FetchBtn');
FetchBtn.addEventListener('click', XHR);
function XHR() {
```

```

let xhr = new XMLHttpRequest();
xhr.open('GET', 'http://dummy.restapiexample.com/api/v1/employees', true);
xhr.getResponseHeader('Content-type', 'application/json');
/*xhr.onprogress = function(){ console.log('currently in progress'); }*/
xhr.onload = function () {
  if (this.status === 200) {
    let resp = JSON.parse(this.responseText);
    console.log(resp);
    let list = document.getElementById('list');

    for (let re in resp.data) {
      let li = `<li>${resp.data[re].employee_name}
${resp.data[re].employee_salary}</li>`;
      list.innerHTML += li;
    }
  }
  xhr.send();
}
let SendBtn = document.getElementById('SendBtn');
SendBtn.addEventListener('click', XHRPOST);
function XHRPOST() {
  let xhr = new XMLHttpRequest();
  xhr.open('POST', 'http://dummy.restapiexample.com/api/v1/create', true)
  xhr.getResponseHeader('Content-type', 'application/json');
  xhr.onprogress = function () {
    console.log('post in progress');
  }
  xhr.onload = function () {
    if (this.status === 200) {
      console.log(this.responseText);
    }
    else {
      console.error('Some error occurred');
    }
  }
  let d = `{ "name": "XHR", "salary": "100000", "age": "23" }`;
  let head = document.getElementById('head');
  head.innerHTML = `<div class="alert alert-success" role="alert">
<strong>Success the data was sent !</strong>
</div>`;
  setTimeout(() => {
    head.innerHTML = ``;
  }, 5000);
  xhr.send(d);
}

```

## • OUTPUT

### After 'GET' request

Fetch content from here

**Fetch.**

- Tiger Nixon 320800
- Garrett Winters 170750
- Ashton Cox 86000
- Cedric Kelly 433060
- Airi Satou 162700
- Brielle Williamson 372000
- Herrod Chandler 137500
- Rhona Davidson 327900
- Colleen Hurst 205500
- Sonya Frost 103600
- Jena Gaines 90560
- Quinn Flynn 342000
- Charde Marshall 470600
- Haley Kennedy 137500
- Tatiana Fitzpatrick 385750
- Michael Silva 198500
- Paul Byrd 725000
- Gloria Little 237500
- Bradley Greer 132000
- Dai Rios 217500
- Jenette Caldwell 345000
- Yuri Berry 675000
- Caesar Vance 106450
- Doris Wilder 85600

**Send**

Live reload enabled.

Unchecked runtime.lastError: The message port closed before a response was received.

Failed to load resource: the server responded with a status of 404 (Not Found)

Object

```
data: Array(24)
  0: {id: "1", employee_name: "Tiger Nixon", employee_salary: "320800", employee_age: "61", prof...
  1: {id: "2", employee_name: "Garrett Winters", employee_salary: "170750", employee_age: "65", ...
  2: {id: "3", employee_name: "Ashton Cox", employee_salary: "86000", employee_age: "66", profil...
  3: {id: "4", employee_name: "Cedric Kelly", employee_salary: "433060", employee_age: "22", pro...
  4: {id: "5", employee_name: "Airi Satou", employee_salary: "162700", employee_age: "33", profil...
  5: {id: "6", employee_name: "Brielle Williamson", employee_salary: "372000", employee_age: "41...
  6: {id: "7", employee_name: "Herrod Chandler", employee_salary: "137500", employee_age: "59", ...
  7: {id: "8", employee_name: "Rhona Davidson", employee_salary: "327900", employee_age: "58", p...
  8: {id: "9", employee_name: "Colleen Hurst", employee_salary: "205500", employee_age: "59", p...
  9: {id: "10", employee_name: "Sonya Frost", employee_salary: "103600", employee_age: "23", pro...
  10: {id: "11", employee_name: "Jena Gaines", employee_salary: "90560", employee_age: "38", pro...
  11: {id: "12", employee_name: "Quinn Flynn", employee_salary: "342000", employee_age: "22", p...
  12: {id: "13", employee_name: "Charde Marshall", employee_salary: "470600", employee_age: "36"...
  13: {id: "14", employee_name: "Haley Kennedy", employee_salary: "137500", employee_age: "43", ...
  14: {id: "15", employee_name: "Tatiana Fitzpatrick", employee_salary: "385750", employee_age: ...
  15: {id: "16", employee_name: "Michael Silva", employee_salary: "198500", employee_age: "66", ...
  16: {id: "17", employee_name: "Paul Byrd", employee_salary: "725000", employee_age: "64", prof...
  17: {id: "18", employee_name: "Gloria Little", employee_salary: "237500", employee_age: "58", ...
  18: {id: "19", employee_name: "Bradley Greer", employee_salary: "132000", employee_age: "41", ...
  19: {id: "20", employee_name: "Dai Rios", employee_salary: "217500", employee_age: "35", profil...
  20: {id: "21", employee_name: "Jenette Caldwell", employee_salary: "345000", employee_age: "38...
  21: {id: "22", employee_name: "Yuri Berry", employee_salary: "675000", employee_age: "40", pro...
  22: {id: "23", employee_name: "Caesar Vance", employee_salary: "106450", employee_age: "21", p...
  23: {id: "24", employee_name: "Doris Wilder", employee_salary: "85600", employee_age: "23", p...
  length: 24
  __proto__: Array(0)
  status: "success"
  __proto__: Object
```

### After POST request

Success the data was sent !

Fetch content from here

**Fetch.**

- Tiger Nixon 320800
- Garrett Winters 170750
- Ashton Cox 86000
- Cedric Kelly 433060
- Airi Satou 162700
- Brielle Williamson 372000
- Herrod Chandler 137500
- Rhona Davidson 327900
- Colleen Hurst 205500
- Sonya Frost 103600
- Jena Gaines 90560
- Quinn Flynn 342000
- Charde Marshall 470600
- Haley Kennedy 137500
- Tatiana Fitzpatrick 385750
- Michael Silva 198500
- Paul Byrd 725000
- Gloria Little 237500
- Bradley Greer 132000
- Dai Rios 217500
- Jenette Caldwell 345000
- Yuri Berry 675000
- Caesar Vance 106450
- Doris Wilder 85600

**Send**

Live reload enabled.

Unchecked runtime.lastError: The message port closed before a response was received.

Failed to load resource: the server responded with a status of 404 (Not Found)

Object

```
data: Array(24)
  0: {id: "1", employee_name: "Tiger Nixon", employee_salary: "320800", employee_age: "61", prof...
  1: {id: "2", employee_name: "Garrett Winters", employee_salary: "170750", employee_age: "65", ...
  2: {id: "3", employee_name: "Ashton Cox", employee_salary: "86000", employee_age: "66", profil...
  3: {id: "4", employee_name: "Cedric Kelly", employee_salary: "433060", employee_age: "22", pro...
  4: {id: "5", employee_name: "Airi Satou", employee_salary: "162700", employee_age: "33", profil...
  5: {id: "6", employee_name: "Brielle Williamson", employee_salary: "372000", employee_age: "41...
  6: {id: "7", employee_name: "Herrod Chandler", employee_salary: "137500", employee_age: "59", ...
  7: {id: "8", employee_name: "Rhona Davidson", employee_salary: "327900", employee_age: "58", p...
  8: {id: "9", employee_name: "Colleen Hurst", employee_salary: "205500", employee_age: "59", p...
  9: {id: "10", employee_name: "Sonya Frost", employee_salary: "103600", employee_age: "23", pro...
  10: {id: "11", employee_name: "Jena Gaines", employee_salary: "90560", employee_age: "38", pro...
  11: {id: "12", employee_name: "Quinn Flynn", employee_salary: "342000", employee_age: "22", p...
  12: {id: "13", employee_name: "Charde Marshall", employee_salary: "470600", employee_age: "36"...
  13: {id: "14", employee_name: "Haley Kennedy", employee_salary: "137500", employee_age: "43", ...
  14: {id: "15", employee_name: "Tatiana Fitzpatrick", employee_salary: "385750", employee_age: ...
  15: {id: "16", employee_name: "Michael Silva", employee_salary: "198500", employee_age: "66", ...
  16: {id: "17", employee_name: "Paul Byrd", employee_salary: "725000", employee_age: "64", prof...
  17: {id: "18", employee_name: "Gloria Little", employee_salary: "237500", employee_age: "58", ...
  18: {id: "19", employee_name: "Bradley Greer", employee_salary: "132000", employee_age: "41", ...
  19: {id: "20", employee_name: "Dai Rios", employee_salary: "217500", employee_age: "35", profil...
  20: {id: "21", employee_name: "Jenette Caldwell", employee_salary: "345000", employee_age: "38...
  21: {id: "22", employee_name: "Yuri Berry", employee_salary: "675000", employee_age: "40", pro...
  22: {id: "23", employee_name: "Caesar Vance", employee_salary: "106450", employee_age: "21", p...
  23: {id: "24", employee_name: "Doris Wilder", employee_salary: "85600", employee_age: "23", p...
  length: 24
  __proto__: Array(0)
  status: "success"
  __proto__: Object

post in progress
{"status":"success","data":{"name":"BOB","salary":"100000","age":"23","id":"68"}}
post in progress
{"status":"success","data":{"name":"BOB","salary":"100000","age":"23","id":"511"}}

```

- **FINDINGS AND LEARNINGS :** We implemented an HTTP request, response and entity header. Here we learned how to use AJAX for implementing request , response and entity headers and we also learned how to use a dummy server for the same.

## EXPERIMENT 5

- **AIM :** Posting messages to the server via electronic mail
- **DESCRIPTION :** E-mail or electronic mail is one of the most popular applications of the Internet. With e-mail a user can send/receive messages from other users on different networks provided he has access to the Internet, thus simplifying the communication system. It is fast, cheap, reliable and convenient mode of communication.

- **CODE**

```
import smtplib, ssl

port = 465 # For SSL

smtp_server = "smtp.gmail.com"

sender_email = "xyz@gmail.com" # Enter your address

receiver_email = "abc@gmail.com" # Enter receiver address

password = input("Type your password and press enter: ")

message = """\
Subject: Konichiwa

This message is sent from Python as an assignment"""

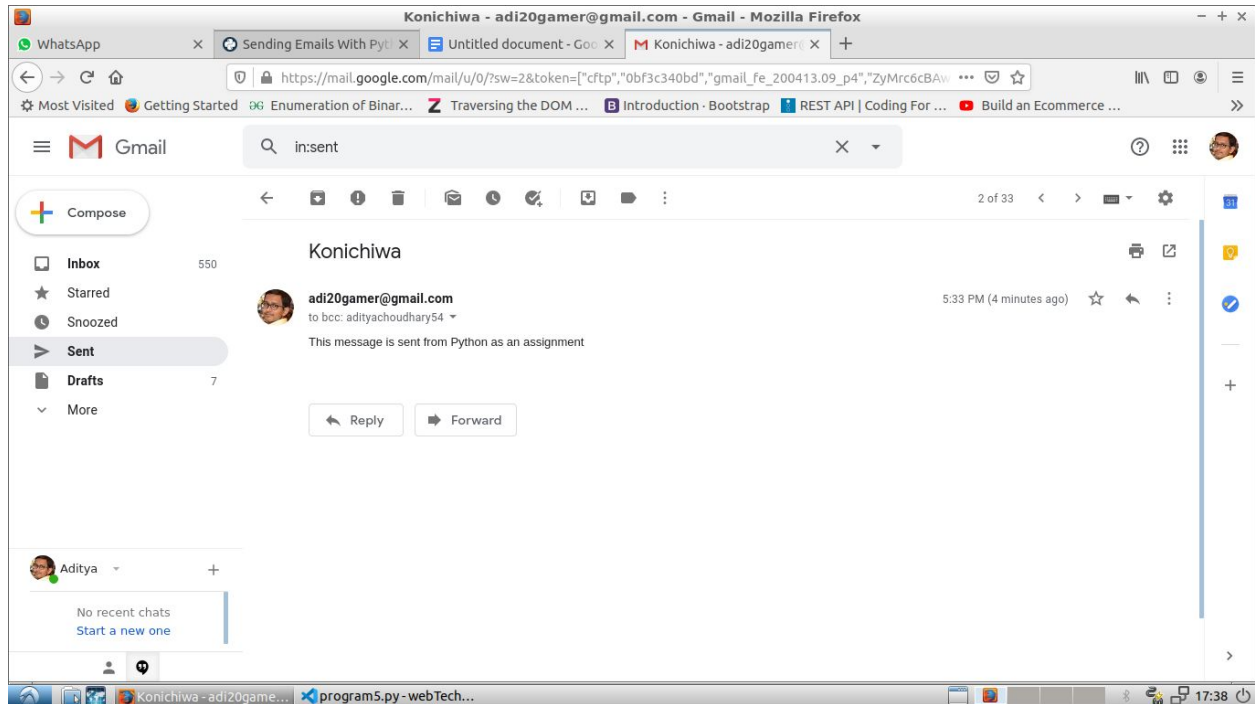
context = ssl.create_default_context()

with smtplib.SMTP_SSL(smtp_server, port, context=context) as server:

    server.login(sender_email, password)

    server.sendmail(sender_email, receiver_email, message)
```

- **OUTPUT**



- **FINDINGS AND LEARNINGS :** From this experiment we learnt about Electronic mail and how it is used. We also learnt how messages can be posted to the server through means of Email. We also learnt to send simple text mails using python and how SMTP and SSL work in conjunction.



## EXPERIMENT 6

- **AIM :** Posting messages to server via SMTP
- **DESCRIPTION :** SMTP(simple mail transfer protocol) is a standard protocol on the Internet for mail transfer. It also provides for control of messages. Another new protocol is ESTMP in place of SMTP. It permits pipelining of the messages that transmission of multiple message commands simultaneously becomes possible. It has checkpoints within messages.

Here we will be posting messages to the server using SMTP and SSL. We try to post messages from one email id to another using a python script. SMTP helps us to send the email and SSL (Secure Sockets Layer) manages the security and ensures that message is transferred safely.

- **CODE**

```
import smtplib
from getpass import getpass
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication

server=smtplib.SMTP('smtp.gmail.com',587)
server.ehlo()
server.starttls()
msg=MIMEMultipart()

sender_add=input("Your email : ")
password=getpass()
server.login(sender_add,password)

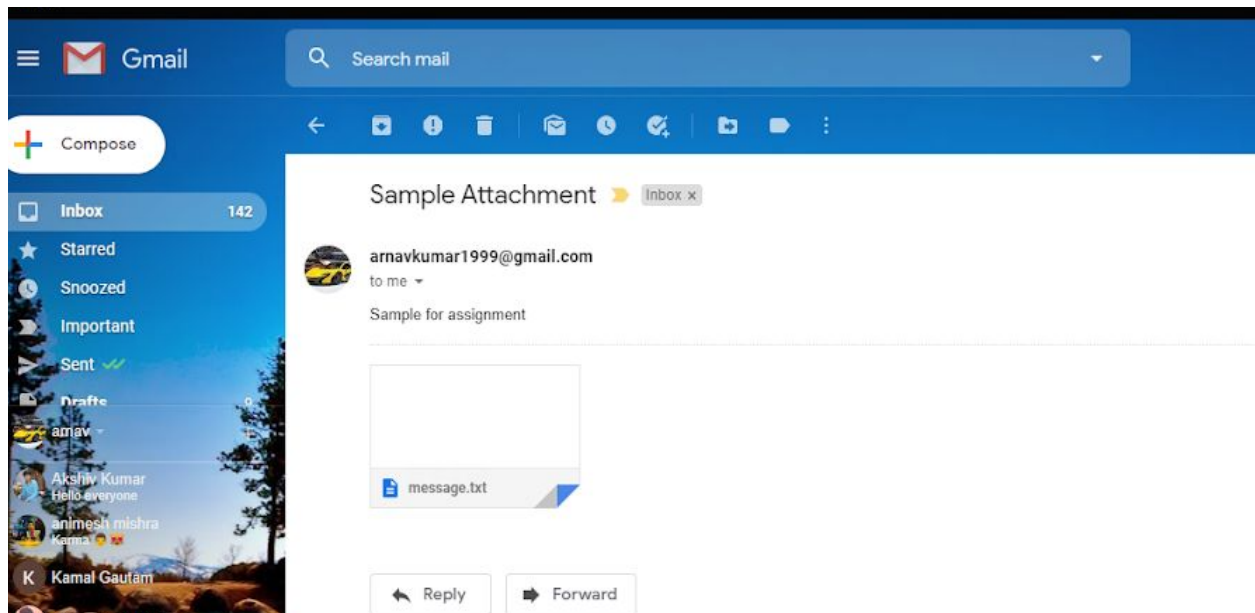
msg['From']=sender_add
msg['To']=input("Receiver's email : ")
msg['Subject']=input("Subject : ")
text=input("Message-body : ")
msg.attach(MIMEText(text))

with open('message.txt', 'rb') as f:
    part=MIMEApplication(f.read())
    part.add_header('Content-Disposition','attachment', filename= 'message.txt')
    msg.attach(part)
server.sendmail(msg['From'],msg['To'], msg.as_string())
```

- **OUTPUT:**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS E:\ARNAV DTU\Semester 6\Web Technology> python -u "e:\ARNAV DTU\Semester 6\Web Technology\attachment_mail.py"
Your email : arnavkumar1999@gmail.com
Password:
Recievers email : arnavkumar1999@gmail.com
Subject : Sample Attachment
Message-body : Sample for assignment
PS E:\ARNAV DTU\Semester 6\Web Technology> |
```



- **FINDINGS AND LEARNINGS :** From this experiment we learnt the usage of SMTP along with SSL. We also learnt how SMTP works in detail and how it can be used to send messages (text, images, videos, files etc) to the server by importing the SMTP module in a python script and then executing it in SSL.

## EXPERIMENT 7

- **AIM**

Design a webpage for enrollment of students form which accepts complete details of students.

- **DESCRIPTION :** A HTML form is used to collect user input. The user input can then be sent to a server for processing. The HTML <form> element defines a form that is used to collect user input. An HTML form contains form elements. Form elements are different types of input elements. The <input> element is the most important form element. The <input> element is displayed in several ways, depending on the type attribute.

- <input type="text"> defines a single-line input field for text input.
- The <label> tag defines a label for many form elements. The <label> element is useful for screen-reader users, because the screen-reader will read out the label.
- The for attribute of the <label> tag should be equal to the id attribute of the <input>
- Radio buttons let a user select ONE of a limited number of choices.
- Check boxes let a user select multiple of a limited number of choices.
- Submit button defines a button for submitting the form data to a form-handler.
- The form-handler is a page on the server with a script for processing input data.
- The action attribute defines the action to be performed when the form is submitted. Usually, the form data is sent to a page on the server when the user clicks on the submit button.
- The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.
- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.

- **CODE**

```
<html>
<head>
<title>Student Registration Form</title>
</head>
<body>
<h3>STUDENT REGISTRATION FORM</h3>
<table align="center" cellpadding = "10">
<form action="register.php" method="post">
<tr>
<td>FIRST NAME</td>
<td><input type="text" name="First_Name" maxlength="30"/> (max 30 characters )</td>
</tr>
<tr>
<td>LAST NAME</td>
<td><input type="text" name="Last_Name" maxlength="30"/> </td>
```

```

</tr>
<tr>
<td>DATE OF BIRTH</td>
<td><input type="date" name="DOB"></td>
</tr>
<tr>
<td>EMAIL ID</td>
<td><input type="email" name="Email_Id" maxlength="100" /></td>
</tr>
<tr>
<td>MOBILE NUMBER</td>
<td><input type="text" name="Mobile_Number" maxlength="10" />(10 digit number) </td>
</tr>
<tr>
<td>GENDER</td>
<td>Male <input type="radio" name="Gender" value="Male" />
      Female <input type="radio" name="Gender" value="Female" /> </td>
</tr>
<tr>
<td>ADDRESS <br /><br /><br /></td>
<td><textarea name="Address" rows="4" cols="30"></textarea></td>
</tr>
<tr>
<td>CITY</td>
<td><input type="text" name="City" maxlength="30" /> (max 30 characters a-z and A-Z) </td>
</tr>
<tr>
<td>PIN CODE</td>
<td><input type="text" name="Pin_Code" maxlength="6" /> (6 digit number)</td>
</tr>
<tr>
<td>STATE</td>
<td><input type="text" name="State" maxlength="30" /> (max 30 characters a-z and A-Z) </td>
</tr>
<tr>
<td>COUNTRY</td>
<td><input type="text" name="Country" value="India" readonly="readonly" /></td>
</tr>
<tr>

```

HOBBIES <input type="checkbox"/> Drawing <input type="checkbox"/> Singing <input type="checkbox"/> Dancing <input type="checkbox"/> Sketching <input type="checkbox"/> Others <input max="30" type="text" value="Other_Hobby"/>
---

Sl.No.	Examination	Board	Percentage	Year of Passing
1	Class X	<input max="30" type="text" value="ClassX_Board"/>	<input max="30" type="text" value="ClassX_Percentage"/>	<input max="30" type="text" value="ClassX_YrOfPassing"/>
2	Class XII	<input max="30" type="text" value="ClassXII_Board"/>	<input max="30" type="text" value="ClassXII_Percentage"/>	<input max="30" type="text" value="ClassXII_YrOfPassing"/>

3
---

```

<td>Graduation</td>
<td><input type="text" name="Graduation_Board" maxlength="30" /></td>
<td><input type="text" name="Graduation_Percentage" maxlength="30" /></td>
<td><input type="text" name="Graduation_YrOfPassing" maxlength="30" /></td>
</tr>
<tr>
<td>4</td>
<td>Masters</td>
<td><input type="text" name="Masters_Board" maxlength="30" /></td>
<td><input type="text" name="Masters_Percentage" maxlength="30" /></td>
<td><input type="text" name="Masters_YrOfPassing" maxlength="30" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td align="center">(10 char max)</td>
<td align="center">(upto 2 decimal)</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>COURSES<br />APPLIED FOR</td>
<td>BCA <input type="radio" name="Course_BCA" value="BCA">
      B.Sc <input type="radio" name="Course_BSc" value="B.Sc">
      B.Tech <input type="radio" name="Course_BTech" value="B.Tech">
</td>
</tr>
<tr>
<td colspan="2">
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</body>
</html>

```

- **OUTPUT**

## STUDENT REGISTRATION FORM

FIRST NAME	<input style="width: 100%;" type="text"/>	(max 30 characters a-z and A-Z)
LAST NAME	<input style="width: 100%;" type="text"/>	(max 30 characters a-z and A-Z)
DATE OF BIRTH	Day: <input style="width: 30px;" type="text"/> Month: <input style="width: 30px;" type="text"/> Year: <input style="width: 30px;" type="text"/>	
EMAIL ID	<input style="width: 100%;" type="text"/>	
MOBILE NUMBER	<input style="width: 100%;" type="text"/>	(10 digit number)
GENDER	Male <input type="radio"/> Female <input type="radio"/>	
ADDRESS	<div style="border: 1px solid black; height: 40px; width: 100%;"></div>	
CITY	<input style="width: 100%;" type="text"/>	(max 30 characters a-z and A-Z)
PIN CODE	<input style="width: 100%;" type="text"/>	(6 digit number)
STATE	<input style="width: 100%;" type="text"/>	(max 30 characters a-z and A-Z)

COUNTRY	<input style="width: 100%;" type="text" value="India"/>																									
HOBBIES	Drawing <input type="checkbox"/> Singing <input type="checkbox"/> Dancing <input type="checkbox"/> Sketching <input type="checkbox"/> Others <input type="checkbox"/> <input style="width: 100px;" type="text"/>																									
QUALIFICATION	<table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th style="width: 5%;">Sl.No.</th> <th style="width: 25%;">Examination</th> <th style="width: 30%;">Board</th> <th style="width: 20%;">Percentage</th> <th style="width: 20%;">Year of Passing</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Class X</td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> </tr> <tr> <td>2</td> <td>Class XII</td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> </tr> <tr> <td>3</td> <td>Graduation</td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> </tr> <tr> <td>4</td> <td>Masters</td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> <td><input style="width: 100%;" type="text"/></td> </tr> </tbody> </table> <div style="display: flex; justify-content: space-around; font-size: x-small; margin-top: -10px;"> <span>(10 char max)</span> <span>(upto 2 decimal)</span> </div>	Sl.No.	Examination	Board	Percentage	Year of Passing	1	Class X	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	2	Class XII	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	3	Graduation	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	4	Masters	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Sl.No.	Examination	Board	Percentage	Year of Passing																						
1	Class X	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																						
2	Class XII	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																						
3	Graduation	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																						
4	Masters	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																						
COURSES APPLIED FOR	BCA <input type="radio"/> B.Com <input type="radio"/> B.Sc <input type="radio"/> B.A <input type="radio"/>																									

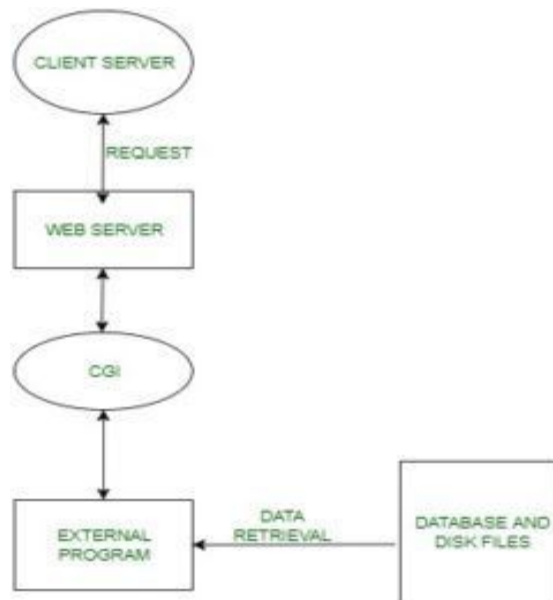
- **FINDINGS AND LEARNINGS :** From this experiment we learnt about the HTML forms. We learnt about how forms work and how they are created in HTML and php. We also created a student registration form which is often required during enrollment in a particular university. We also learnt about action , target and method attributes which are a part of form handler.

## EXPERIMENT 8

- **AIM**

To implement CGI for transacting with database at client and server system

- **DESCRIPTION :** Common Gateway Interface (also known as CGI) is not a kind of language but just a specification(set of rules) that helps to establish a dynamic interaction between a web application and the browser (or the client application). The CGI programs make possible communication between client and web servers. Whenever the client browser sends a request to the webserver the CGI programs send the output back to the web server based on the input provided by the client-server
  - CGI is the standard for programs to interface with HTTP servers.
  - CGI programming is written dynamically generating web pages that respond to user input or webpages that interact with software on the server



- **CODE:**

**Client**

```
<html>
<head>
  <title>Player Transaction</title>
</head>
<body>
  <h3>Select The Option to Transact with database</h3>
  <ol>
    <li>Add a player</li>
    <li>Retrieve All players</li>
    <li>Find a player</li>
  </ol>
```



```
<form action="" method="post" id="frm">
<input type="text" name="topt" id="topt">
<input type="submit" value="Submit Data">
</form>
```

```
<div id="add"></div>
```

```
<script>
```

```
let formCode=`<h1>Add A player</h1>
```

```
<h3><p>Information Required.</p></h3>
```

```
<form action="/cgi-bin/script1.pl" method="Post">
```

```
<table>
```

```
<tr>
```

```
<td>Name:</td>
```

```
<td><input type="text" name="name"><td>
```

```
</tr>
```

```
<tr>
```

```
<td>Gender:</td>
```

```
<td><select name="gender" size="1">
```

```
<option>Female</option>
```

```
<option>Male</option>
```

```
<option>Transgender</option>
```

```
</select></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Nationality:</td>
```

```
<td><input type="text" name="profession"><td>
```

```
</tr>
```

```
<tr>
```

```
<td>Sports:</td>
```

```
<td><input type="checkbox" name="sport"
value="Cricket">Cricket
```

```
<input type="checkbox" name="sport"
value="Hockey">Hockey
```

```
<input type="checkbox" name="sport"
value="TableTennis">TableTennis
```

```
<input type="checkbox" name="sport"
value="Football">Football</td>
```

```
<input type="text" name="opt" value="1" hidden >
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2"><input type="submit"></td>
```

```
</tr>
```

```
</table>
```

```
</form>`;
```

```
let e=document.getElementById('frm');
```

```

e.addEventListener('submit',(e)=>{
    e.preventDefault();
    // console.log(document.getElementById('topt').value);
    if(document.getElementById('topt').value==1 &&
    document.getElementById('add').innerHTML.length==0){
        document.getElementById('add').innerHTML+=formCode;
    }
    if(document.getElementById('topt').value==2&&
    document.getElementById('add').innerHTML.length==0){
        document.getElementById('add').innerHTML+=`
        <form action="/cgi-bin/script1.pl" method="Post" id='getAll'>
            <input type="text" name="opt" value="2" hidden
></form>`;
        document.getElementById('getAll').submit();
    }
    if(document.getElementById('topt').value==3&&
    document.getElementById('add').innerHTML.length==0){
        document.getElementById('add').innerHTML+=`
        <form action="/cgi-bin/script1.pl" method="Post" id='getAll'>
            <p>You Have selected to Find a player</p>
            Enter Name of Player<input type="text" name="delete"
id="del" value="">
            <input type="text" name="opt" value="3" hidden    >
            <input type="submit">
        </form>`;
    }
});
</script>
</body>
</html>

```

### CGI (server)

```
#!/D:\Drive\xampp\perl\bin\perl.exe"
```

```
use strict;
```

```
use CGI 'standard';
```

```
my $opt=param('opt');
```

```
my $list;
```

```
print header,
```

```
start_html(),
```

```
end_html;
```

```

#Adding To server
if($opt eq 1)
{

    my $name = param('name');
    my $gender = param('gender');
    my $profession = param('profession');
    my @sports = param('sport');

    if (@sports)
    {
        $list = join ' ', @sports;
    }
    else
    {
        $list = 'Null';
    }

    open (FILE, ">> file3.txt");
    print FILE "$name,$gender,$profession,$list";
    print FILE "\n";
    close FILE;
    print "<h4>Successfully Submitted Data</h4>";
}

if($opt eq 2){
print "<h2>list of all Players</h2>";
# print "Content-Type: text/plain; charset=UTF-8\n\n";
my $filename = 'file3.txt';
open(my $fh, '<:encoding(UTF-8)', $filename)
    or die "Could not open file '$filename' $!";

while (my $row = <$fh>) {
    chomp $row;
    # print($row);
    my @spl=split(',',$row);

    print('<h3>',@spl[0],</h3>');
    print('<table>');
    print ("<tr><td>Name:</td><td>", @spl[0] ,"</td></tr>");
    print ("<tr><td>Gender:</td><td>", @spl[1] ,"</td></tr>");
    print ("<tr><td>Nationality:</td><td>", @spl[2] ,"</td></tr>");
    print ("<tr><td>Sports:</td><td>", @spl[3] ,"</td></tr>");
}

```

```

print('</table>');
print('<br>');
# h4 td($row);

}
}

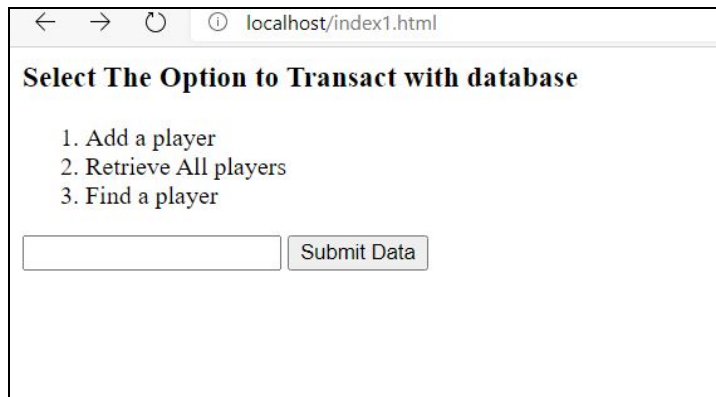
if($opt eq 3){
    my $name=param('delete');
    # print "$name";
    my $filename = 'file3.txt';
    open(my $fh, '<:encoding(UTF-8)', $filename)
    or die "Could not open file '$filename' $!";
    my $found=0;
    while (my $row = <$fh>) {
        chomp $row;
        # print($row);
        my @spl=split(',',$row);
        if($name eq @spl[0] ){

            print('<h5>,@spl[0], Found, Delails Are</h5>');
            print('<table>');
            print ("<tr><td>Name:</td><td>", @spl[0] , "</td></tr>");
            print ("<tr><td>Gender:</td><td>", @spl[1] , "</td></tr>");
            print ("<tr><td>Nationality:</td><td>", @spl[2] , "</td></tr>");
            print ("<tr><td>Sports:</td><td>", @spl[3] , "</td></tr>");

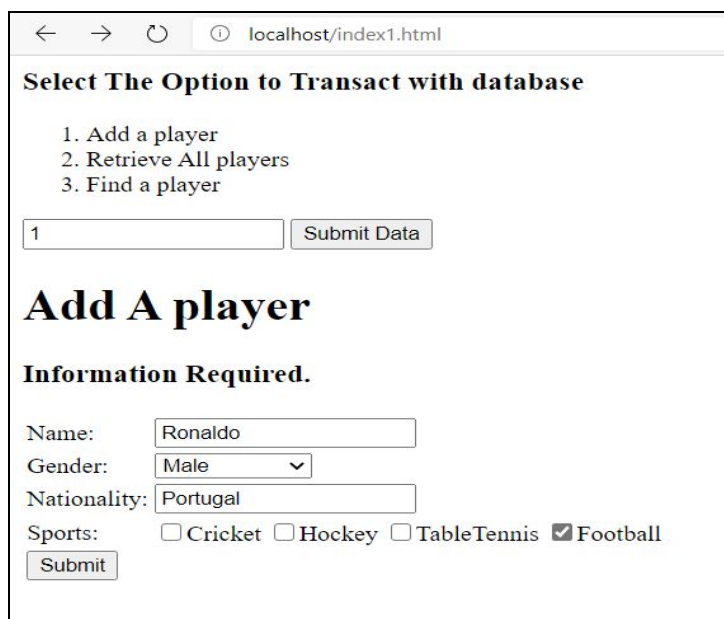
            print('</table>');
            $found=1;
        }
    }
    if($found eq 0){
        print "Name Not Found";
    }
}
# print "done\n";

```

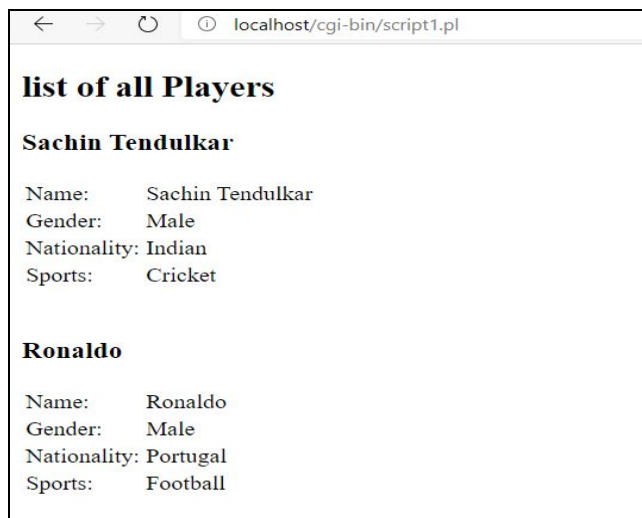
- **OUTPUT**



A screenshot of a web browser window with the address bar showing 'localhost/index1.html'. The page has a title 'Select The Option to Transact with database'. Below the title is a numbered list: '1. Add a player', '2. Retrieve All players', and '3. Find a player'. At the bottom, there is an empty text input field and a 'Submit Data' button.



A screenshot of the same web browser window. The text input field now contains the number '1'. Below the 'Submit Data' button, the page title has changed to 'Add A player'. Under the title is the heading 'Information Required.'. This is followed by form fields: 'Name:' with the value 'Ronaldo', 'Gender:' with a dropdown menu showing 'Male', and 'Nationality:' with the value 'Portugal'. Below these is a 'Sports:' section with four checkboxes: 'Cricket', 'Hockey', 'TableTennis', and 'Football'. The 'Football' checkbox is checked. A 'Submit' button is at the bottom.



A screenshot of a web browser window with the address bar showing 'localhost/cgi-bin/script1.pl'. The page title is 'list of all Players'. The content is divided into two sections. The first section is titled 'Sachin Tendulkar' and lists his details: Name: Sachin Tendulkar, Gender: Male, Nationality: Indian, and Sports: Cricket. The second section is titled 'Ronaldo' and lists his details: Name: Ronaldo, Gender: Male, Nationality: Portugal, and Sports: Football.

← → ↻ ⓘ localhost/index1.html

**Select The Option to Transact with database**

1. Add a player
2. Retrieve All players
3. Find a player

You Have selected to Find a player

Enter Name of Player:

**Sachin Tendulkar Found, Details Are**

Name: Sachin Tendulkar  
Gender: Male  
Nationality: Indian  
Sports: Cricket

- **FINDINGS AND LEARNINGS :** From this experiment we learnt about the Common gateway interface. We also learnt it's principles, and how the working takes place at the client side and the server side respectively. We used PERL for the making the CGI program because it is the most preferred and oldest language with respect to CGI. We also made a program that interacts with both client and the database at XAMPP server.
-