

action = "http://localhost:9090/examples/serulets/servlet/
AssetB2">

<h2> Select an option : </h2>

Painting <input type = "radio" value = "Painting"
name = "options">

drawing <input type = "radio" value = "drawing"
name = "options">

singing <input type = "radio" value = "singing"
name = "options">

swimming <input type = "radio" value = "swimming"
name = "options">

<input type = "reset" value = "Reset" style = "background-
color: red; color: white">

<input type = "submit" value = "Submit" style = "background-color: red; color: white">

</form>

</body>

</html>

C

frontend
08/10/2024

Java script



* Set A

- 1] Write a program to make use of following JSP implicit objects:
- i) out: To display current Date and Time.
 - ii) request: To get header information.
 - iii) response: To Add cookie
 - iv) config: get the parameters value defined in `<init-param>`
 - v) application: get the parameter value defined in `<context-param>`
 - vi) session: display Current Session ID
 - vii) pageContext: To set and get the attributes.
 - viii) page: get the name of Generated Servlet.

→ AssesetA1.jsp

```

<%@ page language="java" import="java.util.*"%>
<html><Head><title> AssesetA1 </title></Head>
<body> <H2> Welcome
<BR> 1) Current Time is:
<% out.println(new Date()); %>
<BR> 2) Request User-Agent :
<% = request.getHeader("User-Agent")%>
<BR> 3) response.addCookie (new Cookie ("class",
"ITYBCS"));
out.print ("cookie Added"); 
<BR> 4) City init param value:

```

```

<% config . getInitParameter ("city") %>
<BR> 5) State context parameter :
<% = application . getInitParameter ("state") %>
<BR> 6) User session Id : <% = session . getId () %>
<BR> 7) Page context attribute :
<% pageContext . setAttribute ("Test", "Test value"); %>
<BR> 8) Generated Servlet Name :
<% = page . getClass () . getName () %>
<H2> </body> <HTML>

```

- 2) Create a JSP page which will accept the file extension and display all files in the current directory having that extension. Each filename should appear as a hyperlink on screen.

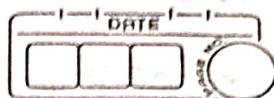
→ Ass6Az.jsp

```

<% page language ="java" import ="java.io.*" %>
<html>
<body>
<center>
<h3>
<% String extension = request . getParameter ("txts");
if (extension != null && ! extension . isEmpty ())
{
    out . println ("Entered File Extension = " + extension );
    File dir = new File (application . getRealPath ("/") +
        "aaa");
}

```

EXPERIMENT : No.



```
if (dirs.exists() && dirs.isDirectory())
{
    String[] files = dirs.list();
    boolean fileFound = false;
    for (String file : files)
        if (file.endsWith("." + extension))
            fileFound = true;
    out.println("<br><a href = \"aaa/\" + file +
               \">" + file + "</a>\"");
}
if (!fileFound)
{
    out.println("<br>No files found with the
               extension: ." + extension);
}
else
{
    out.println("<br>Please enter a file
               extension .");
}
out.println("<br>Directory 'aaa' does not exist!);
```

Teacher's Sign.: _____

```
%>
</h3>
<center>
<body>
<html>
```

* Set B

- 5] Create a JSP page, which accepts user name in a text box and greets the user according to the time on server side. Example: User name : ABC
Output: Good morning ABC / Good Afternoon ABC / Good Evening ABC

→ Ass4SetB1.jsp

```
<%@page language="java" import="java.util.*"%>
<html>
<head>
<title> Ass4SetB1 </title>
</head>
<body>
<H1>
<%
String name = request.getParameter("t1");
out.println("Welcome "+name); %>
```

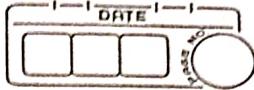
✓

```
Calendar cal = Calendar.getInstance();
int h = cal.get(Calendar.HOUR_OF_DAY);
```

```
if(h>=1 && h<=11)
    out.println("<BR> Good Morning....");
```

EXPERIMENT :

No.



```
else if (h >= 12 & h <= 16)
    out.println("<BR> Good Afternoon ...");
else if (h >= 17 || h <= 19)
    out.println("<BR> Good evening ...");
else
    out.println("<BR> Good Night ...");
/>
</body>
</html>
```

'C'

~~frontend~~
08/10/2024



Teacher's Sign.:

Set A

- 1) Write a client - server program which displays the server machine's date and time on the client machine.

client.java

```
import java.net.*;  
import java.io.*;  
import java.util.*;
```

```
class Ass7SetAclient
```

```
{
```

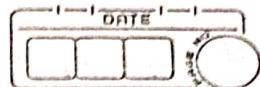
```
    public static void main (String args []) throws  
        Host  
        unknownHostException, IOException.
```

```
    Socket s = new socket ("localhost", 5000),  
    InputStream is = s.getInputStream(),  
    DataInputStream dis = new DataInputStream(is);  
    String str = dis.readUTF();  
    System.out.println (str);
```

```
}
```

server.java

```
import java.net.*;
import java.io.*;
import java.util.*;
public class AsynchronousServer extends Thread {
    public void run() {
        try {
            ServerSocket ss = new ServerSocket(1000);
            System.out.println("Server started, waiting for client");
            Socket s = ss.accept();
            System.out.println("Client connected");
            OutputStream os = s.getOutputStream();
            DataOutputStream dos = new DataOutputStream(os);
            dos.writeUTF("Server Date = " + new Date());
        } catch (Exception e) {
        }
    }
}
```



- 2) Write a program which sends the name of text file from the client to server & displays the contents of the file on the client machine if the file is not found display an error message.

→ Client.java

```
import java.net.*;
import java.io.*;
class AssetA2client {
    public static void main (String args[]) throws
        UnknownHostException, IOException {
        Socket s = new Socket ("localhost", 5000);
        BufferedReader br = new BufferedReader (new
            InputStreamReader (System.in));
        OutputStream os = s.getOutputStream ();
        DataOutputStream dos = new DataOutputStream (os);
        InputStream is = s.getInputStream ();
        DataInputStream dis = new DataInputStream (is);
        System.out.println ("Enter file name to search");
        String fn = br.readLine ();
        dos.writeUTF (fn);
        String msg = dis.readUTF ();
        System.out.println (msg);
    }
}
```

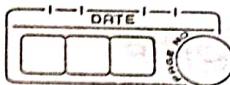
Server.java

```
import java.net.*;
import java.io.*;

class Ass7Server {
    public static void main (String args []) throws
        UnknownHostException, IOException
    {
        ServerSocket ss = new ServerSocket (5000);
        System.out.println ("Server started, waiting for file to search");
        Socket s = ss.accept ();
        System.out.println ("Client connected");
        InputStream is = s.getInputStream ();
        DataInputStream dis = new DataInputStream (is);
        OutputStream os = s.getOutputStream ();
        DataOutputStream dos = new DataOutputStream (os);
        String fn = dis.readUTF ();
        File f1 = new File (fn);
        String msg = "";
        int c;
        if (f1.exists ())
        {
            System.out.println ("File exists");
            FileInputStream fis = new FileInputStream (f1);
            msg = "file contents are : \n";
            while ((c = fis.read ()) != -1)
                msg = msg + (char) c;
        }
    }
}
```

EXPERIMENT :

No.



dos::writeUTF("msg");
fis::close();

↓

else

dos::writeUTF("File not Found");

dis::close();

dos::close();

s::close();

↓

↓

program terminates with error file not found
and exit status 1 because of an error.

exit

if we want to read file from disk then we have to use read function.

when we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

if we want to read file from disk then we have to use read function.

Teacher's Sign.:

* Set 8

83] Write a program to accept a list of file names on the client machine & check how many exist on the server. Display appropriate message on the client side.

→ client.java.

```
import java.net.*;
import java.io.*;

class Ass7Set8Client
{
    public static void main (String args[])
        throws
            UnknownHostException , IOException
    {
        try
        {
            socket s = new socket ("localhost" , 5000),
            BufferedReader br = new BufferedReader (new InputStreamReader
                (System.in));
            OutputStream os = s.getOutputStream (),
            DataOutputStream dos = new DataOutputStream (os),
            InputStream is = s.getInputStream (),
            DataInputStream dis = new DataInputStream (is);
            while (true)
            {
                System.out.println ("Enter file name to search:");
                String fn = br.readLine ();
                dos.writeUTF (fn);
                dos.flush ();
                String ans = dis.readUTF ();
                if (ans.equals ("File not found"))
                    System.out.println ("File not found");
                else
                    System.out.println ("File found");
            }
        }
    }
}
```

```
socket s = ss.accept();
System.out.println ("Client connected : ");
InputStream is = s.getInputStream(),
DataInputStream dis = new DataInputStream(is);

OutputStream os = s.getOutputStream();
DataOutputStream dos = new DataOutputStream(os);

ArrayList al = new ArrayList();

while (true)
{
    fn = dis.readUTF();
    if (fn.equals ("ENO") || fn.equals ("end"))
        break;
    else
        al.add (fn);
    System.out.println ("file name received : " + fn);
}

for (int i=0; i<al.size(); i++)
{
    fi = new File (al.get (i).toString ());
    if (!fi.exists ())
        dos.writeUTF (al.get (i).toString () + " is
exist.");
    else
}
```

`dos.writeUTF(cal.get(i).toString() + " dot is exist");`

dis.close();

`dos.close();` close the output stream to the file

c.close(); // close the connection

1