

CS8661
INTERNET
PROGRAMMING
LABORATORY

LAB MANUAL

VI Semester

2019 – 2020

OBJECTIVES:

- To be familiar with Web page design using HTML/XML and style sheets
- To be exposed to creation of user interfaces using Java frames and applets.
- To learn to create dynamic web pages using server side scripting.
- To learn to write Client Server applications.
- To be familiar with the PHP programming.
- To be exposed to creating applications with AJAX

LIST OF EXPERIMENTS

1. Create a web page with the following using HTML
 - a. To embed a map in a web page
 - b. To fix the hot spots in that map
 - c. Show all the related information when the hot spots are clicked.
2. Create a web page with the following.
 - a. Cascading style sheets.
 - b. Embedded style sheets.
 - c. Inline style sheets. Use our college information for the web pages.
3. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
4. Write programs in Java using Servlets:
 - i. To invoke servlets from HTML forms
 - ii. Session tracking using hidden form fields and Session tracking for a hit count
5. Write programs in Java to create three-tier applications using servlets for conducting on- line examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.
6. Install TOMCAT web server. Convert the static web pages of programs into dynamic web pages using servlets (or JSP) and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
7. Redo the previous task using JSP by converting the static web pages into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database.
8. Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document
9. Validate the form using PHP regular expression.
 - ii. PHP stores a form data into database.
10. Write a web service for finding what people think by asking 500 people's opinion for any consumer product.

TOTAL: 60PERIODS

OUTCOMES: Upon Completion of the course, the students will be able to:

- CO 1.** Construct Web pages using HTML/XML and style sheets.
- CO 2.** Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.
- CO 3.** Develop dynamic web pages using server side scripting.
- CO 4.** Use PHP programming to develop web applications.
- CO 5.** Construct web applications using AJAX and web services

CBS

Machine learning code with JavaScript

SOFTWARE REQUIRED: Dream Weaver or Equivalent, MySQL or Equivalent, Apache Server, WAMP/XAMPP

Ex No 1:

DATE:

Create a web page with the following using HTML

- a. To embed a map in a web page**
- b. To fix the hot spots in that map**
- c. Show all the related information when the hot spots are clicked**

AIM:

To create a web page which includes a map and display the related information when a hot spot is clicked on the map.

ALGORITHM:

1. Create a html file with map tag.
2. Set the source attribute of the image tag to the location of the image and also set the use map attribute.
3. Specify an area with name, shape and href set to the appropriate values.
4. Repeat step 3 as many hot spots you want to put in the map.
5. Create html files for each and every hot spot the user want to select.

Program

```
<!DOCTYPE html>
<html>

<h3>Mapping an Image

<body>

<p>Click on the different continents of the map to know about them.</p>



<map name="worldmap">
<area shape="rect" coords="184, 36, 272, 158" alt="north america"
      href="https://en.wikipedia.org/wiki/North_America">

<area shape="rect" coords="282, 215, 354, 367" alt="south america"
      href="https://en.wikipedia.org/wiki/South_America">

<area shape="rect" coords="506, 151, 570, 333" alt="africa"
      href="https://en.wikipedia.org/wiki/Africa">

<area shape="rect" coords="618, 42, 791, 162" alt="asia"
      href="https://en.wikipedia.org/wiki/Asia">

<area shape="rect" coords="509, 44, 593, 110" alt="europe"
```

```

href="https://en.wikipedia.org/wiki/Europe">

<area shape="rect" coords="786, 288, 862, 341" alt="australia"
      href="https://en.wikipedia.org/wiki/Australia_(continent)">

<area shape="rect" coords="249, 463, 760, 488" alt="antartica"
      href="https://en.wikipedia.org/wiki/Antarctica">
</map>

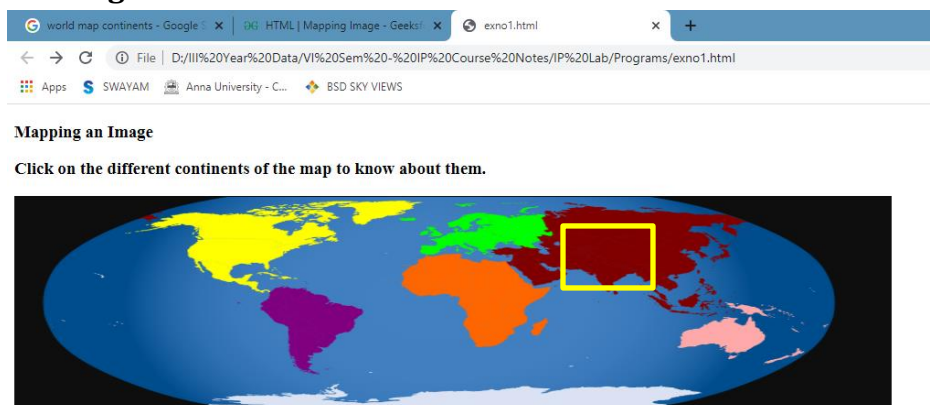
</body>
</html>

```

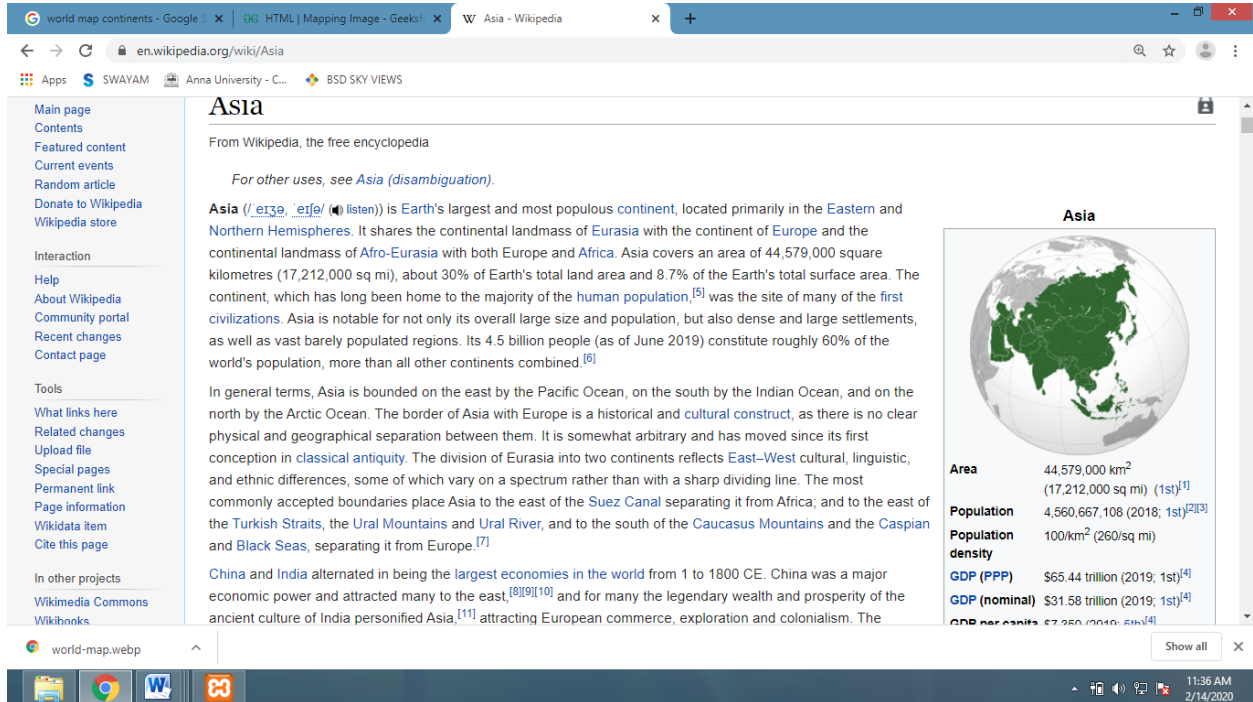
Output



Clicking ASIA



Redirecting to Wikipedia Page



The screenshot shows a web browser window with the Wikipedia page for Asia. The browser's address bar shows the URL `en.wikipedia.org/wiki/Asia`. The page content includes the title "Asia", a brief introduction, and a detailed description of the continent. A map of Asia is displayed on the right side of the page. The sidebar on the left contains various navigation links.

Asia

From Wikipedia, the free encyclopedia

For other uses, see [Asia \(disambiguation\)](#).

Asia (/ˈeɪʒə, ˈeɪʃə/ (listen)) is Earth's largest and most populous *continent*, located primarily in the *Eastern* and *Northern Hemispheres*. It shares the continental landmass of *Eurasia* with the continent of *Europe* and the continental landmass of *Afro-Eurasia* with both *Europe* and *Africa*. Asia covers an area of 44,579,000 square kilometres (17,212,000 sq mi), about 30% of Earth's total land area and 8.7% of the Earth's total surface area. The continent, which has long been home to the majority of the *human population*,^[5] was the site of many of the *first civilizations*. Asia is notable for not only its overall large size and population, but also dense and large settlements, as well as vast barely populated regions. Its 4.5 billion people (as of June 2019) constitute roughly 60% of the world's population, more than all other continents combined.^[6]

In general terms, Asia is bounded on the east by the Pacific Ocean, on the south by the Indian Ocean, and on the north by the Arctic Ocean. The border of Asia with Europe is a historical and *cultural construct*, as there is no clear physical and geographical separation between them. It is somewhat arbitrary and has moved since its first conception in *classical antiquity*. The division of Eurasia into two continents reflects *East–West* cultural, linguistic, and ethnic differences, some of which vary on a spectrum rather than with a sharp dividing line. The most commonly accepted boundaries place Asia to the east of the *Suez Canal* separating it from Africa; and to the east of the *Turkish Straits*, the *Ural Mountains* and *Ural River*, and to the south of the *Caucasus Mountains* and the *Caspian* and *Black Seas*, separating it from Europe.^[7]

China and *India* alternated in being the *largest economies in the world* from 1 to 1800 CE. China was a major economic power and attracted many to the east,^{[8][9][10]} and for many the legendary wealth and prosperity of the ancient culture of India personified Asia,^[11] attracting European commerce, exploration and colonialism. The

Asia

Area 44,579,000 km² (17,212,000 sq mi) (1st)^[1]

Population 4,560,667,108 (2018; 1st)^{[2][3]}

Population density 100/km² (260/sq mi)

GDP (PPP) \$65.44 trillion (2019; 1st)^[4]

GDP (nominal) \$31.58 trillion (2019; 1st)^[4]

GDP per capita \$13,250 (2019; 5th)^[4]

Show all

RESULT: Thus the creation of a web page which includes a map and display the related information when a hot spot is clicked on the map was executed successfully.

EX NO 2

DATE:

Create a web page with the following.

a. Cascading style sheets.

b. Embedded style sheets.

c. Inline style sheets. Use our college information for the web pages.

AIM

To apply inline style sheets to a web page using HTML tags.

ALGORITHM:

1. Start the program with inclusion of <html> tag.
2. The <style> tag is used for combining various styles.
3. Specify the font color, weight and size inside style as its attributes.
4. Close the corresponding tags with a '/' symbol.
5. Stop the program.

Cascading Style Sheet (CSS) is used to set the style in web pages which contain HTML elements. It sets the background color, font-size, font-family, color etc property of elements in web pages.

There are three types of CSS which are given below:

- Inline CSS
- Internal or Embedded CSS
- External CSS

Inline CSS: Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using style attribute.

Program

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline CSS</title>
  </head>

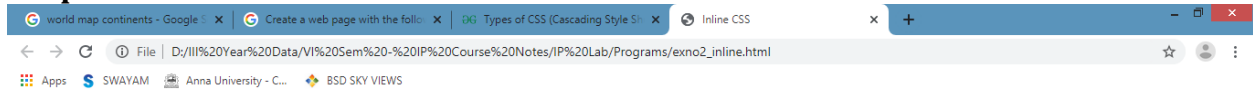
  <body>
    <p style = "color:#009900; font-size:50px;
      font-style:italic; text-align:center;">
      Velammal Engineering College</p>
    <p style = "color:#009940; font-size:50px;
      font-style:bold; text-align:center;">
      Department of Computer Science and Engineering</p>
    <hr>
    <p style = "color:red; font-size:50px;
```

```

        font-style:bold; text-align:center;">
INLINE STYLE SHEET EXAMPLE
    </p>
</body>
</html>

```

Output



Velammal Engineering College

Department of Computer Science and Engineering

INLINE STYLE SHEET EXAMPLE



Internal or Embedded CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Internal CSS</title>
    <style>
      .main {
        text-align:center;
      }
      .GFG {
        color:#009900;
        font-size:50px;
        font-weight:bold;
      }
      .geeks {
        font-style:bold;
        font-size:20px;
      }
    </style>
  </head>
  <body>
    <div class="main">
      <h1>Internal CSS</h1>
    </div>
    <div class="GFG">
      <h2>GeeksforGeeks</h2>
    </div>
    <div class="geeks">
      <h3>GeeksforGeeks</h3>
    </div>
  </body>
</html>

```

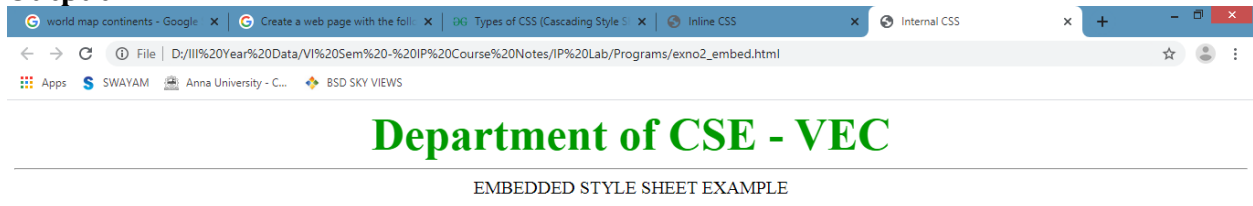


```

    }
  </style>
</head>
<body>
  <div class = "main">
    <div class ="GFG">Department of CSE - VEC</div>
    <hr>
    <div class ="geeks">
      EMBEDDED STYLE SHEET EXAMPLE
    </div>
  </div>
</body>
</html>

```

Output



External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, ... etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

Example: The file given below contains CSS property. This file save with .css extension. For
Ex: exno2_style.css

```

body {
  background-color: powderblue;
}

```

```

.main {
    text-align:center;
}
.GFG {
    color:#009900;
    font-size:50px;
    font-weight:bold;
}
#geeks {
    font-style:bold;
    font-size:20px;
}

```

HTML File

```

<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="exno2_style.css"/>
  </head>

  <body>
    <div class = "main">
      <div class ="GFG">Department of CSE - VEC</div>
      <div id ="geeks">
        External Style Sheet Example
      </div>
    </div>
  </body>
</html>

```

Output



RESULT: Thus the above program to implement style sheets has been executed successfully.

EX NO 3

DATE

Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

AIM

To Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

ALGORITHM

1. Create appropriate html pages for website.
2. Store them in same folder
3. Do necessary validation for web pages

Main.html

```
<html>

  <frameset rows="25%,*">

    <frame src="top.html" name="top"
scrolling = "no" frameborder = "0">

    <frameset cols="25%,75%">

      <frame src="left.html" name="left"
scrolling = "no" frameborder = "0">

      <frame src="right.html" name="right"
scrolling = "auto" frameborder = "0">

    </frameset>

  </frameset>

</html>
```

Top.html

```
<html>

  <body bgcolor="blue">

    <br><br>

    <marquee>

      <h1 align="center"><b><u>ONLINE
BOOK STORAGE</u></b></h1>

    </marquee>

  </body>

</html>
```

Left.html

```
<html>

<body bgcolor="pink">

  <h3>

  <ul>

    <li><a
href="login.html"target="right"><font
color="black"> LOGIN</font></a></li>

    <br><br>

    <li><a href="profile.html"
target="right"><font color="black"> USER
PROFILE</font></a></li>

    <br><br>

    <li><a href="catalog.html"
target="right"><font color="black"> BOOKS
CATALOG</font></a></li>

    <br><br>

    <li><a href="scart.html"
target="right"><font color="black">
SHOPPINGCART</font></a></li>

    <br><br>

    <li><a href="payment.html"
target="right"><font color="black">
PAYMENT</font></a></li>

    <br><br> <br><br>

  </ul>

</body>

</html>
```

Right.html

```
<html>

<body>

  <br><br><br><br><br>

  <h2 align="center">

    <b>

      <p> welcome to online book storage.
Press login if you are having id otherwise
press registration. </p>

    </b>

  </h2>

</body>

</html>
```

Order.html

```
<html>

<head>

  <title>order conformation</title>

</head>

<body bgcolor="cyan">

  <center>

    <h1><b>VEC - CSE Dept</b></h1>

    <pre><strong> <b>Your order Is
Conformed </strong></pre>

    <h2><b>THANK YOU</b></h2>

  </center>

</body>

</html>
```

Login.html

[illegible]

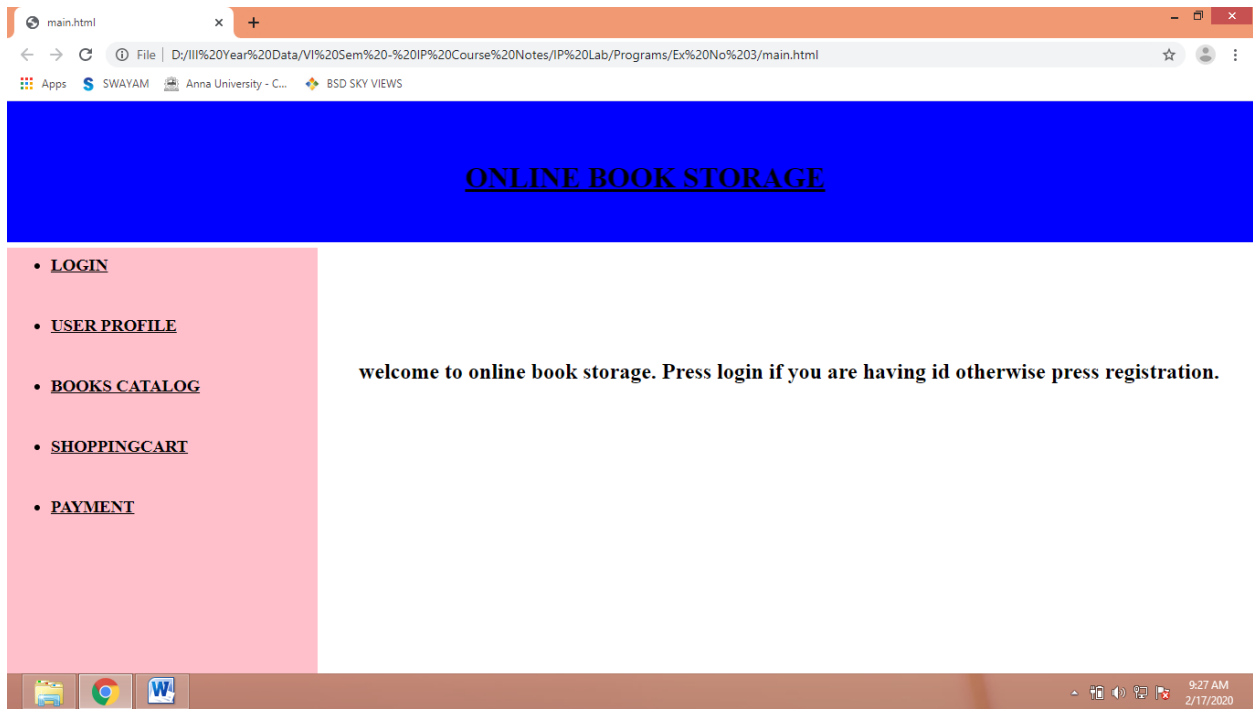
Profile.html

[illegible]

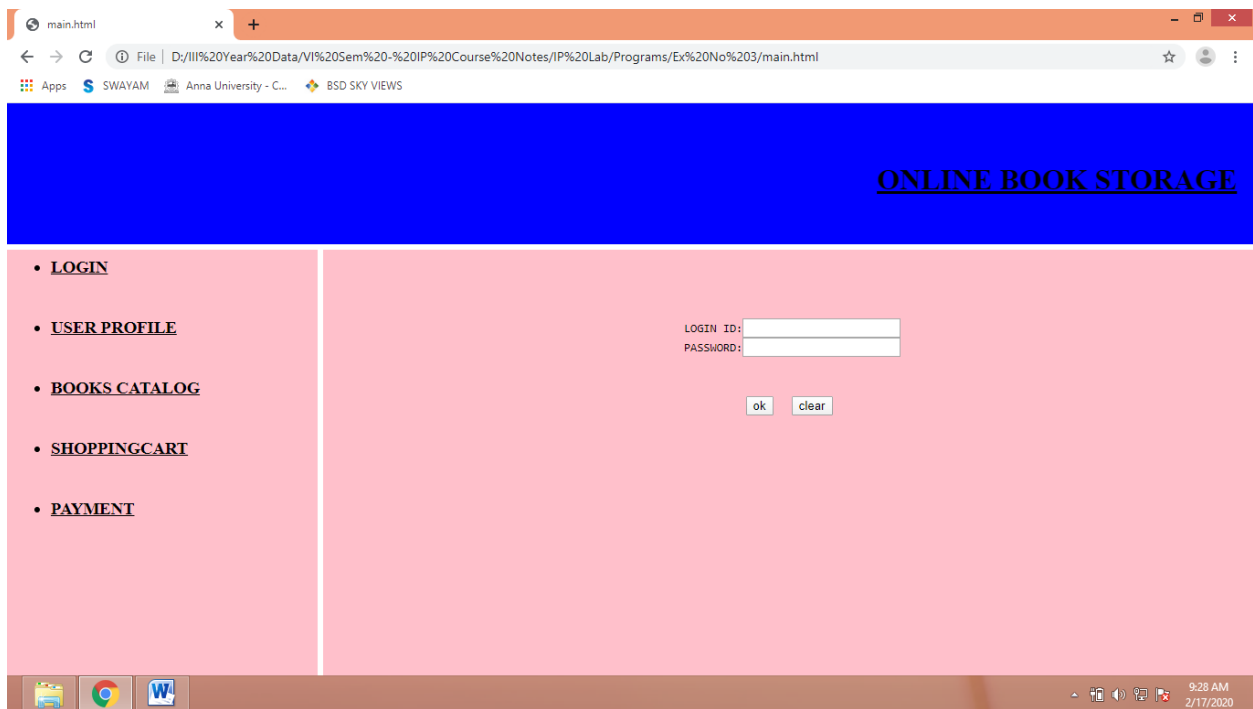
Payment.html

[illegible]

Output



Log in Page



User Profile Page with Validation

main.html

File | D:/III%20Year%20Data/VI%20Sem%20-%20IP%20Course%20Notes/IP%20Lab/Programs/Ex%20No%203/main.html

Apps SWAYAM Anna University - C... BSD SKY VIEWS

This page says
Enter all the details
OK

ONLINE BOOK STORAGE

- [LOGIN](#)
- [USER PROFILE](#)
- [BOOKS CATALOG](#)
- [SHOPPINGCART](#)
- [PAYMENT](#)

NAME :
ADDRESS :
CONTACT NUMBER :
LOGINID :
PASSWORD :

ok clear

9:28 AM
2/17/2020

Books Catalog Page with Validation

main.html

File | D:/III%20Year%20Data/VI%20Sem%20-%20IP%20Course%20Notes/IP%20Lab/Programs/Ex%20No%203/main.html

Apps SWAYAM Anna University - C... BSD SKY VIEWS

This page says
INVALID INPUT
OK

ONLINE BOOK STORAGE

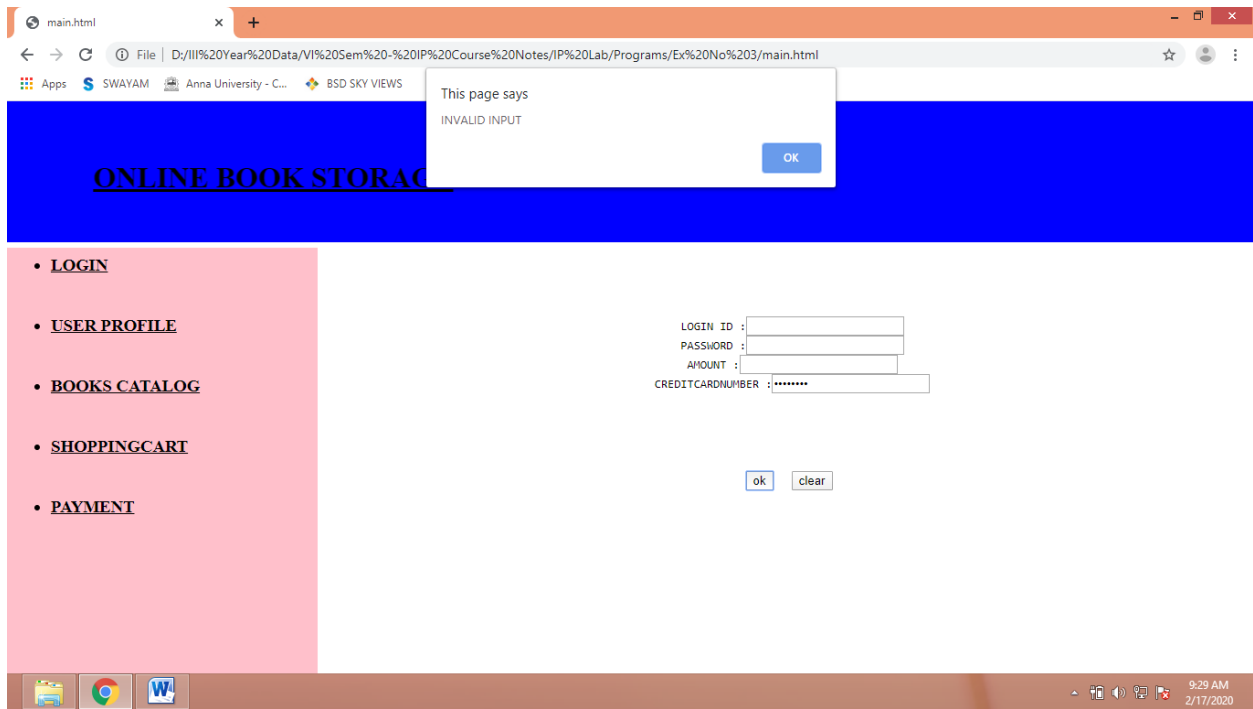
- [LOGIN](#)
- [USER PROFILE](#)
- [BOOKS CATALOG](#)
- [SHOPPINGCART](#)
- [PAYMENT](#)

LOGIN ID :
TITLE :
NO.OF BOOKS :
COST OF BOOK :

ok clear

9:29 AM
2/17/2020

Payment Page with Validation



RESULT: Thus the above program to implement Validation for Registration, User Login, User Profile and Payment has been completed successfully.

Ex No 4:

Date:

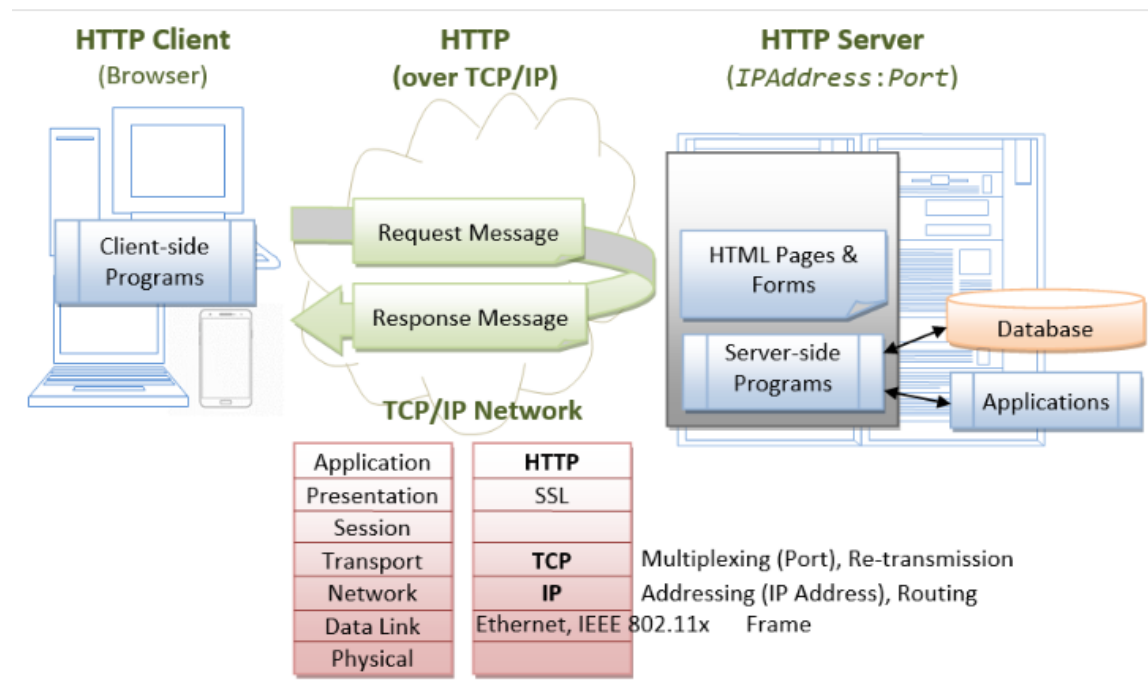
TOMCAT INSTALLATION AND CONFIGURATION OF WEB SERVER TO IMPLEMENT SERVER SIDE PROGRAMMING

Web Application (Webapp)

A web application (or webapp), unlike standalone application, runs over the Internet. Examples of webapps are google, amazon, facebook and twitter.

A webapp is typically a 3-tier (or multi-tier) client-server database application run over the Internet as illustrated in the diagram below. It comprises five components:

1. **HTTP Server:** E.g., Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS), nginx, Google Web Server (GWS), and others.
2. **HTTP Client (or Web Browser):** E.g., Internet Explorer (MSIE), FireFox, Chrome, Safari, and others.
3. **Database:** E.g., Open-source MySQL, PostgreSQL, Apache Derby, mSQL, SQLite, OpenOffice's Base; Commercial Oracle, IBM DB2, SAP SyBase, Microsoft SQL Server, Microsoft Access; and others.
4. **Client-Side Programs:** Could be written in HTML Form, JavaScript, and others.
5. **Server-Side Programs:** Could be written in Java Servlet/JSP, ASP, PHP, Perl, Python, JavaScript, and others.



How to Install Tomcat and Get Started with Java Servlet Programming

STEP 1: Download and Install Tomcat

For Windows

1. Goto <http://tomcat.apache.org> ⇒ Under "Tomcat 9.0.{xx} Released", where {xx} is the latest update number ⇒ Click "Download" ⇒ Under "9.0.{xx}" ⇒ Binary Distributions ⇒ Core ⇒ "zip" (e.g., "apache-tomcat-9.0.{xx}.zip", about 11 MB).
2. UNZIP (right-click ⇒ Extract All) the downloaded file into your project directory "c:\myWebProject". Tomcat shall be unzipped into directory "c:\myWebProject\apache-tomcat-9.0.{xx}".
3. For EASE OF USE, we shall shorten and rename this directory to "c:\myWebProject\tomcat".
4. Take note of Your Tomcat Installed Directory. Hereafter, I shall refer to the Tomcat installed directory as <TOMCAT_HOME>.

STEP 2: Create an Environment Variable JAVA_HOME

For Windows

You need to create an environment variable (system variable available to all applications) called "JAVA_HOME", and set it to your JDK installed directory.

STEP 3: Configure the Tomcat Server

The Tomcat configuration files, in XML format, are located in the "conf" sub-directory of your Tomcat installed directory, e.g. "c:\myWebProject\tomcat\conf" (for Windows) or "~/myWebProject/tomcat/conf" (for macOS). The important configuration files are:

1. server.xml
2. web.xml
3. context.xml

Make a BACKUP of the configuration files before you proceed!!!

Step 3(a) "conf\server.xml" - Set the TCP Port Number

Use a programming text editor (e.g., Sublime Text, Atom) to open the configuration file "server.xml".

The default TCP port number configured in Tomcat is 8080, you may choose any number between 1024 and 65535, which is not used by existing applications. We shall choose 9999 in this article. (For production server, you should use port 80, which is pre-assigned to HTTP server as the default port number.)

Locate the following lines (around Line 69) that define the HTTP connector, and change port="8080" to port="9999".

```
<!-- A "Connector" represents an endpoint by which requests are received
```

and responses are returned. Documentation at :

Java HTTP Connector: /docs/config/http.html

Java AJP Connector: /docs/config/ajp.html

APR (HTTP/AJP) Connector: /docs/apr.html

Define a non-SSL HTTP/1.1 Connector on port 8080

-->

```
<Connector port="9999" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

Step 3(b) "conf\web.xml" - Enable Directory Listing

Again, use a programming text editor to open the configuration file "web.xml".

We shall enable directory listing by changing "listings" from "false" to "true" for the "default" servlet. This is handy for test system, but not for production system for security.

Locate the following lines (around Line 122) that define the "default" servlet; and change the "listings" from "false" to "true".

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Step 3(c) "conf\context.xml" - Enabling Automatic Reload

We shall add the attribute reloadable="true" to the <Context> element to enable automatic reload after code changes. Again, this is handy for test system but not recommended for production, due to the overhead of detecting changes.

Locate the <Context> start element (around Line 19), and change it to <Context reloadable="true">.

```
<Context reloadable="true">
  .....
  .....
</Context>
```

Start Tomcat Server

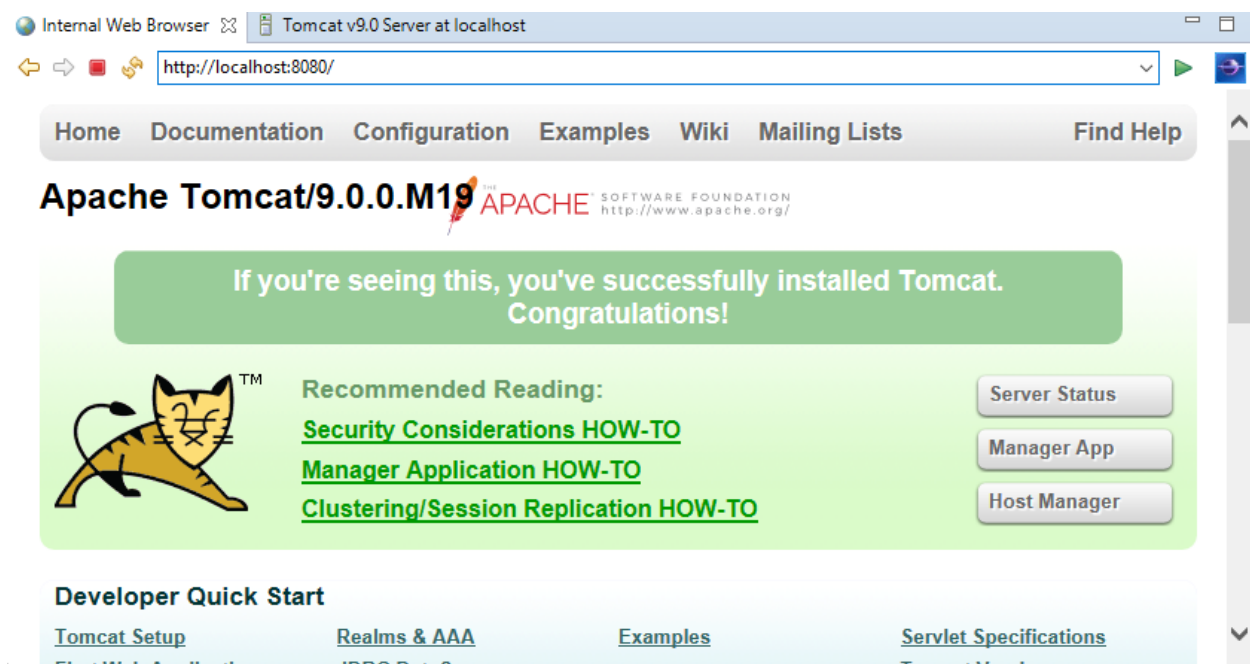
The Tomcat's executable programs and scripts are kept in the "bin" sub-directory of the Tomcat installed directory.

Start Server

For Windows

I shall assume that Tomcat is installed in "c:\myWebProject\tomcat". Launch a CMD shell and issue:

```
c:           // Change drive
cd \myWebProject\tomcat\bin // Change directory to your Tomcat's binary directory
startup     // Run startup.bat to start tomcat server
```



RESULT: Thus the above program to implement Validation for Registration, User Login, User Profile and Payment has been completed successfully.

EX NO: 5a

DATE:

Write programs in Java using Servlets:

i. To invoke servlets from HTML forms

AIM

To write a HTML code which invokes servlet.

ALGORITHM:

1. Start the program.
2. Create a html page which gets two numbers from the user.
3. Use post method to pass the input to the servlet on submit.
4. Create the servlet code to add the values posted by the html.
5. Stop the program.

HTML Code – Login.html

```
<html>

<head>

<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action ="display" method
="post">

name : <input type ="text"
name="name">

password : <input type ="password"
name ="pass">

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

</form>

</body>

</html>
```

Web.xml

```
<servlet>

    <servlet-
name>DisplayDetail</servlet-
name>

    <servlet-
class>Display</servlet-class>

</servlet>

<servlet-mapping>

    <servlet-
name>DisplayDetail</servlet-
name>

    <servlet-
class>/Display</servlet-class>

</servlet-mapping>
```

Servlet Code – Display.java

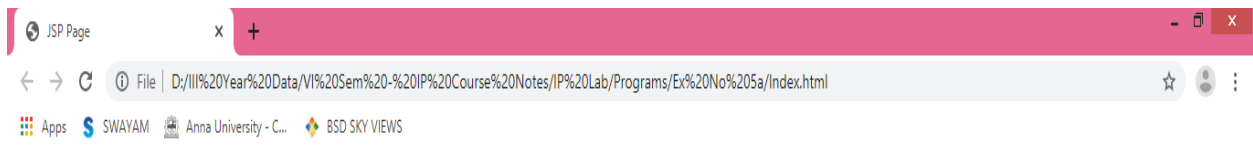
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name="display", urlPatterns={"/display"})
public class display extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String a = request.getParameter("name");
            out.println("<h1>Invoking Servlet using HTML - Output</h1>");
            out.println("<h2>Hi!!!" +a+ "you are now logged in...</h2>");

        } finally {
            out.close();
        }
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}
```


Output



Invoking Servlet using HTML

Name :

Password :



Invoking Servlet using HTML - Output

Hi!!! Shanmughapriya you are now logged in...

RESULT: Thus the above program to to invoke servlets from HTML forms has been completed successfully.

Ex No: 5b

DATE:

Write programs in Java using Servlets:

ii. To hide form fields

AIM

To write a java servlet program for session tracking using hidden form fields.

ALGORITHM

Step 1: Import all the necessary packages.

Step 2: Declare a class FirstServlet1 that extends HttpServlet.

Step 3: In hidden form field a hidden (invisible) textfield is used for maintaining the state of an user.

`<input type='hidden' name='uname' value=" ">`

Step 4: We are storing the name of the user in a hidden textfield and getting that value from another servlet.

Step 5: After entering the name from the HTML page, it goes to the first servlet. First Servlet gets the name and sends as an invisible text field to the servlet2.

Step 6: Second servlet accepts the hidden text field and constructs the invisible text.

HiddenField.htmls

```
<html>
<head><title>Hiding HTML Form
Field - Using Servlet</title></head>
<body>
<h1>Hiding HTML Form Field - Using
Servlet</h1>
<form
action="http://localhost:8080/firstse
rv/FirstServlet1" method="get">
Name:<input type="text"
name="userName"/><br/>
<input type="Submit" value="Hide
the Field"/>
</form>
</body>
</html>
```

Web.xml

```
<web-app>
<servlet>
<servlet-name>SecondServlet1</servlet-
name>
<servlet-class>SecondServlet1</servlet-
class>
</servlet>
<servlet-mapping>
<servlet-name>SecondServlet1</servlet-
name>
<url-pattern>/SecondServlet1</url-
pattern>
</servlet-mapping>
<servlet>
<servlet-name>FirstServlet1</servlet-
name>
<servlet-class>FirstServlet1</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>FirstServlet1</servlet-
name>
<url-pattern>/FirstServlet1</url-pattern>
</servlet-mapping>
</web-app>
```

FirstServlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class FirstServlet1 extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n=request.getParameter("userName");
            out.print("Welcome "+n);
            out.print("<form action='http://localhost:8080/secserv/SecondServlet1' method='get'>");
            out.print("<input type='hidden' name='uname' value='"+n+"'>");
            out.print("<input type='submit' value='go'>");
            out.print("</form>");
            out.close();
        }catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

SecondServlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SecondServlet1 extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    {
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n=request.getParameter("uname");
            out.print("Hello "+n);
            out.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

Output



Text Box - Hidden



Button Hidden



Hiding HTML Form Field - Using Servlet

Name: Shanmughapriya

RESULT: Thus the above program to hide form fields has been completed successfully.

Ex No: 5c

DATE:

Write programs in Java using Servlets:

iii. Session tracking using hidden form fields and Session tracking for a hit count

AIM

To write programs in Java using Servlets to do Session tracking using hidden form fields and Session tracking for a hit count

HTML File

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<body>
<h3>Hit Count Example with HttpSession</h3>

<form method="get" action="http://localhost:8080/MyApp/HitCount">

Click for Hit Count <input type="submit" value="GET HITS">

</form>
</body>
</html>
```

Web. Xml

```
<web-app>
<servlet>
  <servlet-name>HitCount</servlet-name>
  <servlet-class>ServFolder.HitCount</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Serv</servlet-name>
  <url-pattern>/MyApp</url-pattern>
</servlet-mapping>
</web-app>
```

HitCount.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class HitCount extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        HttpSession session = req.getSession();    // this is how to get a session object

        Integer hitNumber = (Integer) session.getAttribute("count"); // retrieving value from
session object

        if(hitNumber == null)
        {
            hitNumber = new Integer(1);
        }
        else
        {
            hitNumber = new Integer(hitNumber.intValue()+1);
        }

        session.setAttribute("count", hitNumber);        // storing the value with session object

        out.println("Your Session ID: " + session.getId()); // never changes in the whole session

        out.println("<br>Session Creation Time: " + new Date(session.getCreationTime()));
// never changes in the whole session

        out.println("<br>Time of Last Access: " + new Date(session.getLastAccessedTime()));
// changes for every hit

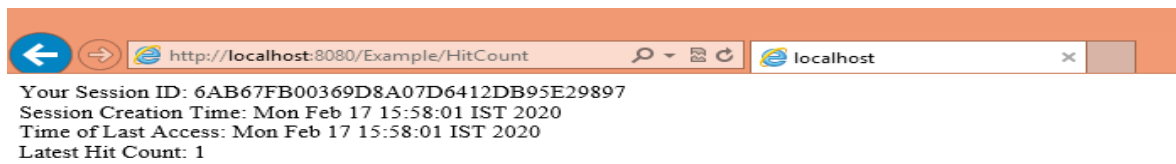
        out.println("<br>Latest Hit Count: " + hitNumber);    // increments by 1 for every hit
    }
}
```

Output



Hit Count Example with HttpSession

Click for Hit Count



After hits – Hit Count and time Changed



RESULT: Thus the above program to do Session tracking for a hit count has been completed successfully.

Ex No 6

DATE

Write programs in Java to create three-tier applications using servlets for conducting on- line examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.

AIM:

To write a java servlet program to conduct online examination and to display student mark list available in a database which has been stored in a database server

ALGORITHM:

Client:

Step 1: In online.html on the client side declare the questions for online exam in which True/False options are displayed that you like to transfer to the server using html form and input type tags.

Step 2: create a submit button and close all the included tags.

Servlet:

Step 1: Import all necessary packages

Step 2: Define a class that extends servlet

Step 3: In the do Post() method, do the following:

- i) Set the content type of the response to "text/html" by using setContentType
- ii) Create a writer to the response
- iii) Get a parameter from the request
- iv) If its value is equal to right answer then add 5 to mark variable
- v) Similarly repeat step for all parameters
- vi) Display the result in an html format using response.getWriter().

Program

online.html

```
<html>
<head>
<title> Database test</title>
</head>
<body>
<center>
<h1> Online Examination System</h1>
<h2> Marks will be saved in Data Base and Result will be shown after the
Exam</h2>
</center>
<form action="http://localhost:8080/exam/exam" method=POST>
<div style="border: solid black 2px;padding:20px">
<b>Enter Your Details</b><br><hr>
<b>Seat number:</b> <input type="text" name="Seat_no"> <br><br>
```

```

<b>Name      :</b> <input type="text" name="Name" size="50">
</div>

<h3>Answer the following Questions</h3><br><hr>
<b>1.Is JAVA a platform independency</b><br>
<input type="radio" name="group1" value="True">True
<input type="radio" name="group1" value="False">False<br> <hr>
<b>2.ASP .NET is a client side programming</b><br>
<input type="radio" name="group2" value="True">True
<input type="radio" name="group2" value="False">False<br> <hr>
<b>3.MATHEMATICS is the backbone of engineering</b><br>
<input type="radio" name="group3" value="True">True
<input type="radio" name="group3" value="False">False<br> <hr>
<b>4.You are working in IBM machines</b><br>
<input type="radio" name="group4" value="True">True
<input type="radio" name="group4" value="False">False<br> <hr>
<b>5.C++ is a fully OOP's language</b><br>
<input type="radio" name="group5" value="True">True
<input type="radio" name="group5" value="False">False<br> <hr>
<center>
<input type="submit" value="Submit to Calculate Marks"><br><br>
</center>
</form>
</body>
</html>

```

Exam.java:

```

import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class exam extends HttpServlet
{
String message,Seat_no,Name,ans1,ans2,ans3,ans4,ans5;
int Total=0;
Connection connect;
Statement stmt =null;
ResultSet rs=null;
public void doPost(HttpServletRequest request,HttpServletResponse response)
throws ServletException,IOException
{
try
{
String url="jdbc:odbc:StudentDB2";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

```

```

connect=DriverManager.getConnection(url," "," ");
message="Connection Successful";
}
catch(ClassNotFoundException cnfex)
{
cnfex.printStackTrace();
}
catch(SQLException sqlex)
{
sqlex.printStackTrace();
}
catch(Exception excp)
{
excp.printStackTrace();
}
Seat_no=request.getParameter("Seat_no");
Name=request.getParameter("Name");
ans1=request.getParameter("group1");
ans2=request.getParameter("group2");
ans3=request.getParameter("group3");
ans4=request.getParameter("group4");
ans5=request.getParameter("group5");
if(ans1.equals("True"))
Total+=2;
if(ans2.equals("False"))
Total+=2;
if(ans3.equals("True"))
Total+=2;
if(ans4.equals("True"))
Total+=2;
if(ans5.equals("False"))
Total+=2;
try
{
Statement stmt=connect.createStatement();
String query="INSERT into StudentTable VALUES (" +Seat_no+", '"+Name+"', '"+Total+"')";
//*****" +Seat_no,Name,Marks"+")" + " VALUES("
int result=stmt.executeUpdate(query);
println("<h1>"+message+"</h1>\n");
out.println("<h3>DataBase Updated");
out.println("<br><br>");
out.println("<b>"+ "The Student Database is as follows");
out.println("<table border=5>");
try
{
Statement stmt=connect.createStatement();

```

```
String query="SELECT * FROM StudentTable";
rs=stmt.executeQuery(query);
```

```
out.println("<th>"+ "Seat_no" + "</th>");
out.println("<th>"+ "Name" + "</th>");
out.println("<th>"+ "Marks" + "</th>");
while(rs.next())
{
out.println("<tr>");
out.println("<td>"+rs.getInt(1)+"</td>");
out.println("<td>"+rs.getString(2)+"</td>");
out.println("<td>"+rs.getInt(3)+"</td>");
out.println("</tr>");
}
out.println("</table>");
}
catch(SQLException ex)
{}
finally
{
try
{
if(rs!=null)
rs.close();
if(stmt!=null)
stmt.close();
if(connect!=null)
connect.close();
}
catch(SQLException e)
{}
}
out.println("<center>");
out.println("<h1>Thanks!</h1>\n");
out.println("</center>");
out.println("</body></html>");
}}
```

Web.xml:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0" metadata-complete="true">
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
```

```

</description>
<servlet>
<servlet-name>exam</servlet-name>
<servlet-class>exam</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>exam</servlet-name>
<url-pattern>/exam</url-pattern>
</servlet-mapping>
</web-app>

```

Execution steps:

1. Create the database StudentDB2.mdb in which Studenttable is created. Note that we have to create an empty database by specifying simply the field names such as Seat_no, Name and Marks.
2. Create the DSN for Student database: Open the Control panel and double click on the Administrative Tools icon. Then double click on the Datasources(ODBC) icon. The window appears as:
3. Click on User DSN tab and select Microsoft Access Driver (*.mdb) and then click on the finish button. Then type the Data Source Name as StudentDB2 and click on select button to select the database file for corresponding DSN. Click on OK button. Now connection to the database is done using JDBC-ODBC driver.
4. Create a html program as online.html in which True/False questions are displayed.
5. Then Create a java program as Exam.java which is a servlet code for computing the total score of each student and it displays the database contents.
6. Compile the java program by using javac exam.java. then class file will be created and place the corresponding class file to the location c:\ "Program Files\apache-tomcat-7.0.29\webapps\exam\WEB-INF\class.
7. Then edit the web.xml file in WEB-INF for servlet name and servlet class as exam
8. Now open online.html file in a browser window the web page opens and if we click on submit button the student information should get updated in the database by invoking a servlet.

Output

Database test

File | D:/III%20Year%20Data/VI%20Sem%20-%20IP%20Course%20Notes/IP%20Lab/Programs/Ex%20No%206/Exam.html

Apps SWAYAM Anna University - C... BSD SKY VIEWS

Online Examination System

Marks will be saved in Data Base and Result will be shown after the Exam

Enter Your Details

Seat number:

Name:

Answer the following Questions

1.Is JAVA a platform independency
☐ True ☐ False

2.ASP.NET is a client side programming
☐ True ☐ False

3.MATHEMATICS is the backbone of engineering
☐ True ☐ False

4.You are working in IBM machines
☐ True ☐ False

5.C++ is a fully OOP's language
☐ True ☐ False

Submit to Calculate Marks

result.html

HTML table border Attribute Tryit Editor v3.6

File | D:/III%20Year%20Data/VI%20Sem%20-%20IP%20Course%20Notes/IP%20Lab/Programs/Ex%20No%206/result.html

Apps SWAYAM Anna University - C... BSD SKY VIEWS

Student Mark Sheet

Seat No	Name	Marks
1001	Priya	95

Result

Thus the java servlet program to conduct online examination and to display student mark list available in a database was executed successfully and output is verified.

Convert the static web pages of programs into dynamic web pages using servlets (or JSP) and cookies. Hint: Users information (user id, password, credit card number) would be stored in Data Base. Each user should have a separate Shopping Cart.

AIM:

To convert the static web pages of programs into dynamic web pages using JSP and cookies.
Hint Given: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

ALGORITHM

1. Create necessary HTML files and Servlets
2. Map the servlets in Web.xml
3. Run the program

Main.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<body bgcolor="orange">
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1><br /><br />
<h2 align="center"><br><hr><br>
<b>Welcome to <b>ONLINE BOOK SHOPPING</b><br><br>
Press LOGIN if you are having <b>LOG IN ID</b><br><br>
Otherwise press <b>REGISTRATION</b> </b></h2><br> <hr>
<br /><br />
<div align=center style="font-size:150%;">
<a href="/tr/login.html">LOGIN</a> <br><br>
<a href="/tr/reg.html"> REGISTRATION</a></div>
</body>
</html>
```

Login.html

```
<html>
<body bgcolor="orange"><br /><br /><br />
<center>
<h1>Log In Page</h1>
<h2>Online Book Shopping</h2>
<h3>Enter Your Crendentials</h3>
<center>
```


Servlets

Login.java

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{ public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{ PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\"");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");
try
{ Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id.equal(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\" /tr/login.html\">press LOGIN to RETRY</a>");
}
else
{
pw.println("VALID LOGIN ID<br><br>");
pw.println("<h3><ul>");
pw.println("<li><a href=\"profile.html\"><fontcolor=\"black\">USER
PROFILE</font> </a></li><br><br>");
pw.println("<li><a href=\"catalog.html\"><fontcolor=\"black\">BOOKS
CATALOG</font></a></li><br><br>");
pw.println("<li><a href=\"order.html\"><fontcolor=\"black\">ORDER
CONFIRMATION</font> </a></li><br><br>");
}
```

```

}
pw.println("</body></html>");
}
catch(Exception e)
{ resp.sendError(500,e.toString());
}
}
Login.java
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{ public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
    PrintWriter pw=resp.getWriter();
    pw.println("<html><body bgcolor=\"pink\"");
    String name=req.getParameter("name");
    String addr=req.getParameter("addr");
    String phno=req.getParameter("phno");
    String id=req.getParameter("id");
    String pwd=req.getParameter("pwd");
    int no=Integer.parseInt(phno);
    try
    {
        Driver d=new oracle.jdbc.driver.OracleDriver();
        DriverManager.registerDriver(d);
        Connection
        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
        ;
        Statement stmt=con.createStatement();
        String sqlstmt="select id,password from login";
        ResultSet rs=stmt.executeQuery(sqlstmt);
        int flag=0;
        while(rs.next())
        {
            if(id.equal(rs.getString(1))&&pwd.equals(rs.getString(2)))
            {
                flag=1;
            }
        }
        if(flag==1)
        {
            pw.println("SORRY INVALID ID ALREADY EXISTS TRY AGAIN WITH NEW
            ID<br><br>");
        }
    }
}

```

```

pw.println("<a href=\" /tr/reg.html\">press REGISTER to RETRY</a>");
}
else
{ Statement stmt1=con.createStatement();
stmt1.executeUpdate("insertintologin
values(\"+names\",\"+addr\",\"+no\",\"+id\",\"+pwd+)\");
pw.println("YOUR DETAILS ARE ENTERED<br><br>");
pw.println("<a href=\" /tr/login.html\">press LOGIN to login</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{ resp.sendError(500,e.toString());
} }}

```

login.java

```

import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet {
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\"");
String title=req.getParameter("title");
try
{
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
pw.println(",div align=\"center\">");
pw.println("TITLE :"+rs.getString(1)+"<br>");
pw.println("AUTHOR :"+rs.getString(2)+"<br>");
pw.println("VERSION :"+rs.getString(3)+"<br>");
pw.println("PUBLISHER :"+rs.getString(4)+"<br>");
pw.println("COST :"+rs.getString(5)+"<br>");
}
}
}

```

```

pw.println("</div");
flag=1;
}
if(flag==0)
{
pw.println("SORRY INVALID TITLE TRY AGAIN <br><br>");
pw.println("<a href=\" /tr/catalog.html\">press HERE to RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
Login.java
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{ public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{ PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\"");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");
try
{ Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id.equal(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
}
}

```

```

if(flag==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\" /tr/login.html\">press LOGIN to RETRY</a>");
}
else
{
pw.println("VALID LOGIN ID<br><br>");
pw.println("<h3><ul>");
pw.println("<li><ahref=\"profile.html\"><fontcolor=\"black\">USER
PROFILE</font> </a></li><br><br>");
pw.println("<li><ahref=\"catalog.html\"><fontcolor=\"black\">BOOKS
CATALOG</font></a></li><br><br>");
pw.println("<li><ahref=\"order.html\"><fontcolor=\"black\">ORDER
CONFIRMATION</font> </a></li><br><br>");
}
pw.println("</body></html>");
}
catch(Exception e)
{ resp.sendError(500,e.toString());
}
}
Reg.html
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{ public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\"");
String name=req.getParameter("name");
String addr=req.getParameter("addr");
String phno=req.getParameter("phno");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");
int no=Integer.parseInt(phno);
try
{
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")

```

```

;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id.equal(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}}
if(flag==1)
{
pw.println("SORRY INVALID ID ALREADY EXISTS TRY AGAIN WITH NEW
ID<br><br>");
pw.println("<a href=\" /tr/reg.html\">press REGISTER to RETRY</a>");
}
else
{ Statement stmt1=con.createStatement();
stmt1.executeUpdate("insertintologin
values(\"+names\",\"+addr\",\"+no+\",\"+id+\",\"+pwd+)\");
pw.println("YOUR DETAILS ARE ENTERED<br><br>");
pw.println("<a href=\" /tr/login.html\">press LOGIN to login</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{ resp.sendError(500,e.toString());
}}
Catlog.java
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet {
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\"");
String title=req.getParameter("title");
try
{
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);

```

```

Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
pw.println("div align=\"center\">");
pw.println("TITLE :"+rs.getString(1)+"<br>");
pw.println("AUTHOR :"+rs.getString(2)+"<br>");
pw.println("VERSION :"+rs.getString(3)+"<br>");
pw.println("PUBLISHER :"+rs.getString(4)+"<br>");
pw.println("COST :"+rs.getString(5)+"<br>");
pw.println("</div>");
flag=1;
}
if(flag==0)
{
pw.println("SORRY INVALID TITLE TRY AGAIN <br><br>");
pw.println("<a href=\"/tr/catalog.html\">press HERE to RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
}
Profile.java
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{ PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\">");
String id=req.getParameter("id");
try
{

```



```

Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection con=DriverManager.getConnection("jdbc:oracle:thin:
@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select * from login where id="+id+"";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
pw.println("<br><br><br>");
while(rs.next())
{
pw.println("<div align=\"center\">");
pw.println("NAME :"+rs.getString(1)+"<br>");
pw.println("ADDRESS :"+rs.getString(2)+"<br>");
pw.println("PHONE NO :"+rs.getString(3)+"<br>");
pw.println("</div>");
flag=1;
}
if(flag==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\"/tr/profile.html\">press HERE to RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
}
Order.java
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor=\"pink\">");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");

```

```

String title=req.getParameter("title");
String count1=req.getParameter("no"); String date=req.getParameter("date");
String cno=req.getParameter("cno");
int count=Integer.parseInt(count1);
try
{
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
;
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0,amount,x;
while(rs.next())
{
if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\\\"/tr/order.html\\\">press HERE to RETRY</a>");
}
else
{
Statement stmt2=con.createStatement();
String s="select cost from book where title="+title+"";
ResultSet rs1=stmt2.executeQuery(s);
int flag1=0;
while(rs1.next())
{
flag1=1;
x=Integer.parseInt(rs1.getString(1));
amount=count*x;
pw.println("AMOUNT :"+amount+"<br><br><br><br>");
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insertintodetails
values('"+id+"','"+title+"','"+amount+"','"+cno+"')");
pw.println("YOUR ORDER has taken<br>");
}
if(flag1==0)
{

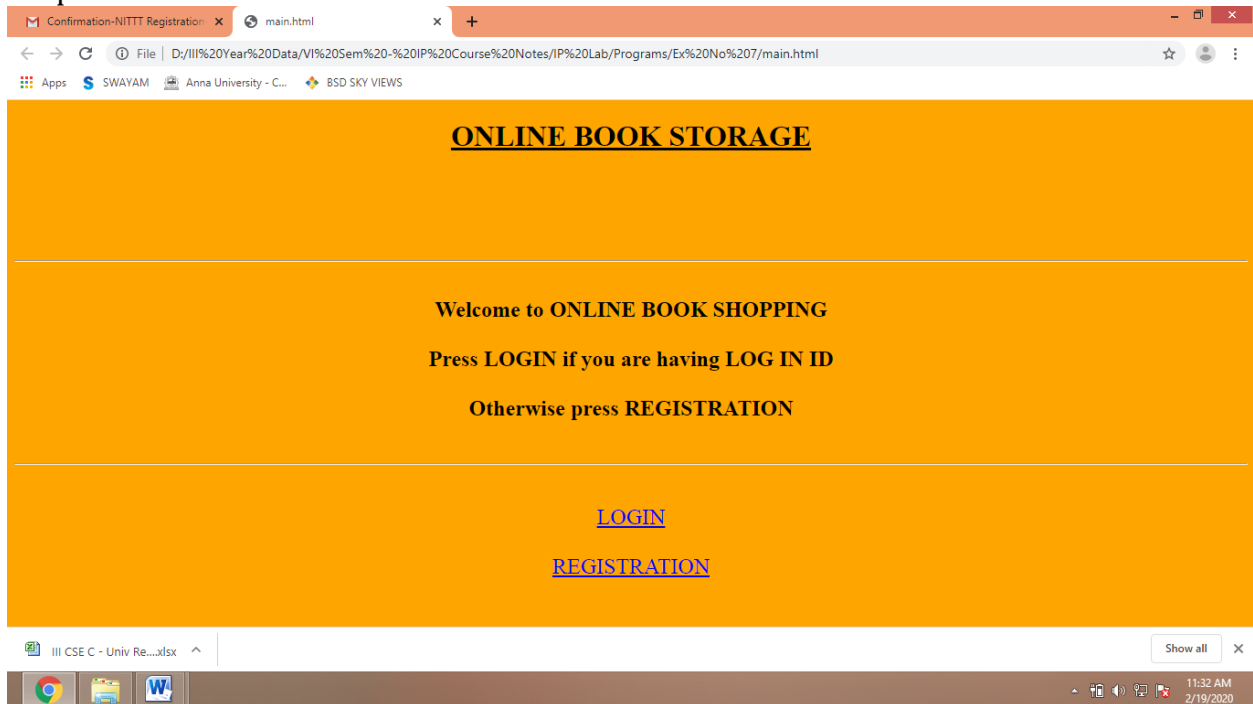
```

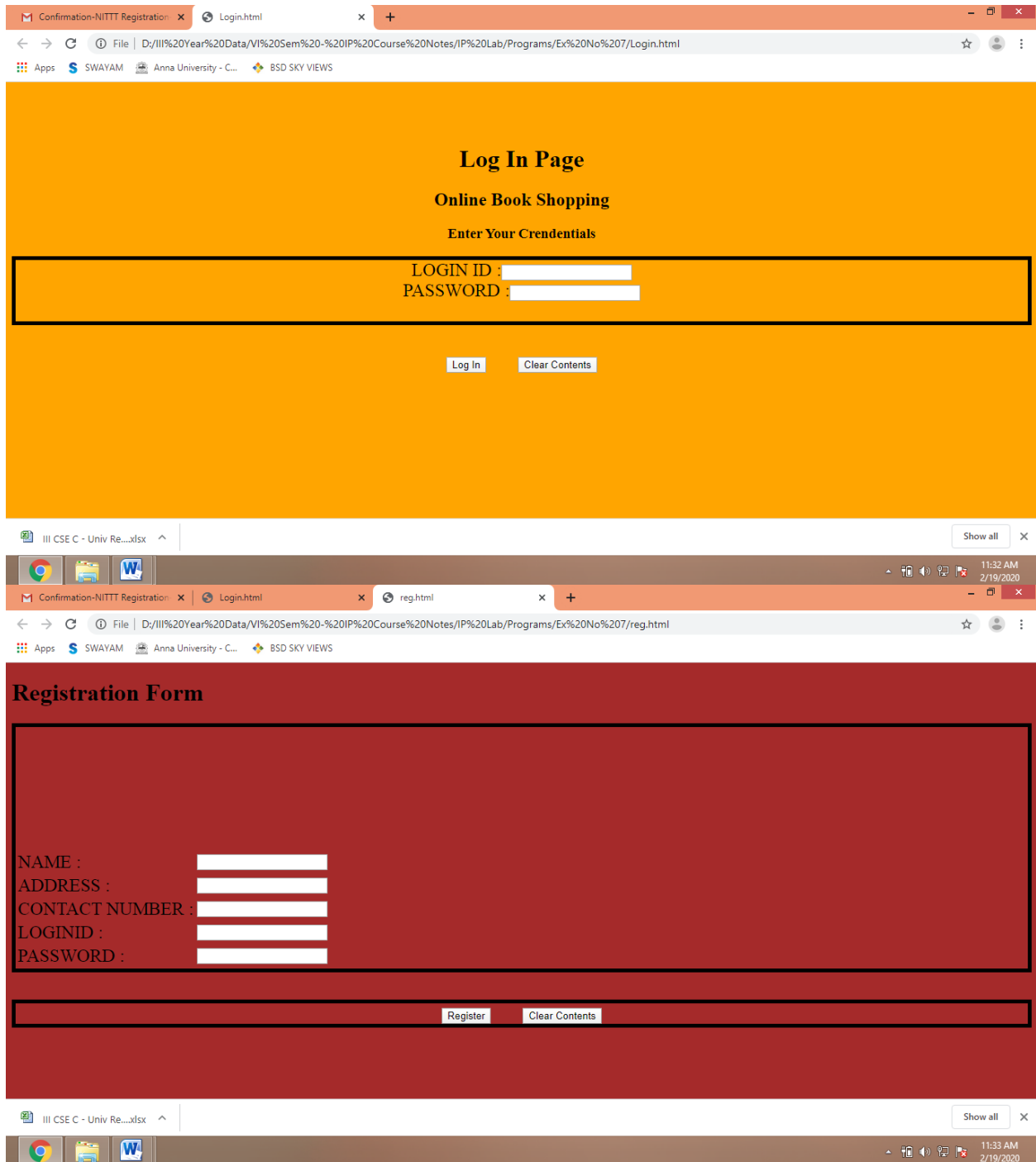
```

pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\\\"/tr/order.html\\\">press HERE to RETRY</a>");
}
}
pw.println("</body></html>");
con.close();
}
catch(Exception e) {
resp.sendError(500,e.toString());
}
}
}

```

Output





Result

Thus the program to convert the static web pages of programs into dynamic web pages using servlets has been completed successfully.

Ex No: 8

DATE

Write a JSP program to implement Calculator functionality such as Addition, Subtraction, multiplication and division

AIM:

To Write a JSP program to implement Calculator functionality such as Addition, Subtraction, multiplication and division

ALGORITHM:

1. Write the JSP Program
2. Implement calculator concepts in the program
3. Configure the server
4. Run the JSP Program in Server

Code

Calculator.jsp

```
<%@ page language="java"%>
<html>
  <head>
    <title>Calculator Using JSP</title>
  </head>
  <body>
    <center>
      <h1>Calculator Using JSP</h1>
      <h2>This Uses Many Form Elements</h2>
    <%
      if(request.getParameter("button") != null) {
    %>
    <% String s = request.getParameter("button");
      int no1,no2;
      if(request.getParameter("no1").equals("")) { no1 = 0; } else no1 =
Integer.valueOf(request.getParameter("no1"));
      if(request.getParameter("no2").equals("")) { no2 = 0; } else no2 =
Integer.valueOf(request.getParameter("no2"));
      if (s.equals("ADD")) {
        out.println("Result is " + (no1 + no2));
      }
      else if (s.equals("SUB")) {
        out.println("Result is " + (no1-no2));
      }
      else if(s.equals("MUL")) {
        out.println("Result is " + (no1*no2));
      }
      else if(s.equals("DIV")) {
```

```

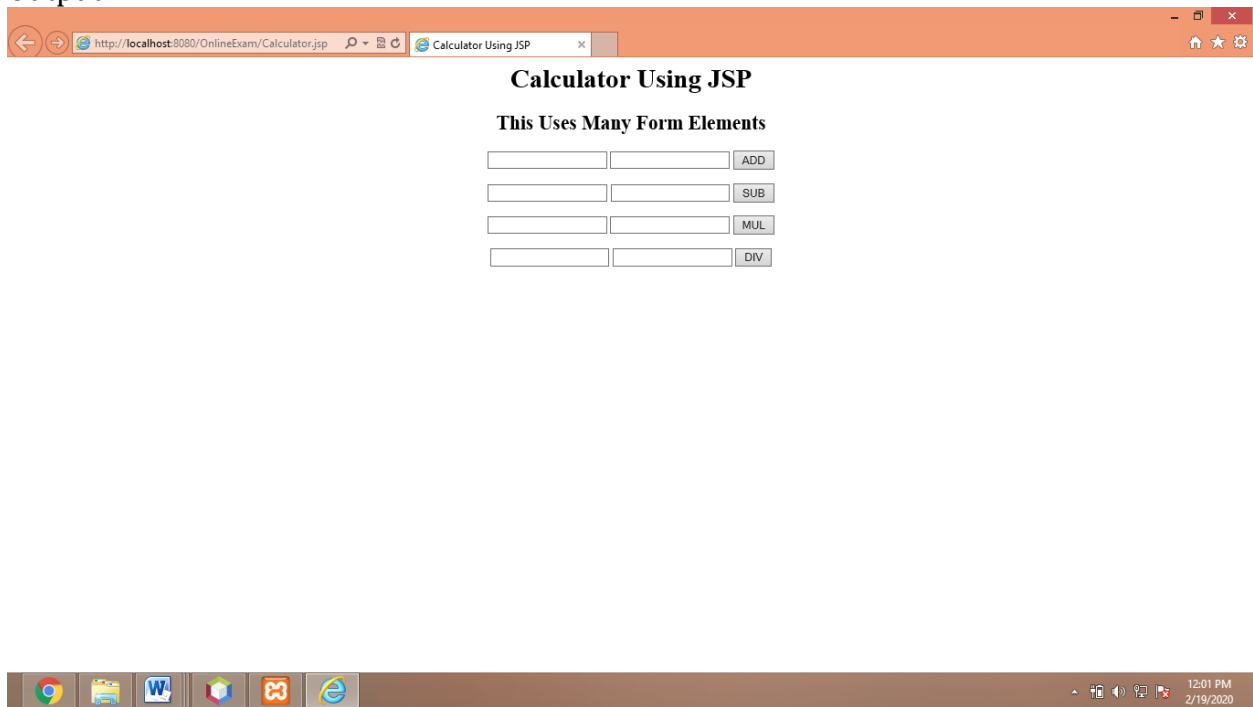
        out.println("Result is " + (no1/no2));
    }

%>
<%
}
%>

<form name="form1" method="get">
    <input name="no1">
    <input name="no2">
    <input type="hidden" name="button" value="ADD">
    <input type="submit" value="ADD">
</form>
<form name="form2" method="get">
    <input name="no1">
    <input name="no2">
    <input type="hidden" name="button" value="SUB">
    <input type="submit" value="SUB">
</form>
<form name="form3" method="get">
    <input name="no1">
    <input name="no2">
    <input type="hidden" name="button" value="MUL">
    <input type="submit" value="MUL">
</form>
<form name="form4" method="get">
    <input name="no1">
    <input name="no2">
    <input type="hidden" name="button" value="DIV">
    <input type="submit" value="DIV">
</form>
</center>
</body>
</html>

```

Output



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/OnlineExam/Calculator.jsp`. The page title is "Calculator Using JSP". Below the title, the text "This Uses Many Form Elements" is displayed. The calculator interface consists of four rows of input fields and operation buttons:

<input type="text"/>	<input type="text"/>	ADD
<input type="text"/>	<input type="text"/>	SUB
<input type="text"/>	<input type="text"/>	MUL
<input type="text"/>	<input type="text"/>	DIV

The Windows taskbar at the bottom shows the time as 12:01 PM on 2/19/2020.

Result

Thus the JSP program to implement Calculator functionality such as Addition, Subtraction, multiplication and division has been completed successfully.

Ex No: 9

DATE

Write a Servlet program to implement the concept of Cookies

AIM

To write a Servlet program to implement the concept of Cookies

ALGORIHTM

1. Write a basic HTML program.
2. Store the values typed/chosen by the user and store it in cookies
3. Retrieve the data from cookies using the servlet program and display it.

Program

CookieIndex.html

```
<!DOCTYPE html>
```

```
<!--
```

To change this license header, choose License Headers in

Project Properties.

To change this template file, choose Tools | Templates
and open the template in the editor.

```
-->
```

```
<html>
```

```
  <head>
```

```
    <title>TODO supply a title</title>
```

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

```
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```

```
  </head>
```

```
  <body>
```

```
    <center>
```

```
      <h1>Understanding Cookies. Storing User Name and  
Password in Cookies</h1>
```

```
      <h2>Login Page</h2>
```

```
      <form
```

```
action="http://localhost:8080/OnlineExam/Serv/Servlet1">
```

```
        User Name:<input type="text"
```

```
name="userName"/><br/>
```

```
        Password:<input type="password"
```

```
name="userPassword"/><br/>
```

```
        <input type="submit" value="Submit to Save in  
Cookies"/>
```

```
      </form>
```



```
</center>
</body>
</html>
```

Servlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Servlet1 extends HttpServlet
{
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) {
        try{
            response.setContentType("text/html");
            PrintWriter pwriter = response.getWriter();

            String name = request.getParameter("userName");
            String password = request.getParameter("userPassword");
            pwriter.print("Hello "+name);
            pwriter.print("Your Password is: "+password);

            //Creating two cookies
            Cookie c1=new Cookie("userName",name);
            Cookie c2=new Cookie("userPassword",password);

            //Adding the cookies to response header
            response.addCookie(c1);
            response.addCookie(c2);
            pwriter.print("<br><a href='http://localhost:8080/Servlet2'>View Details</a>");
            pwriter.close();
        }catch(Exception exp){
            System.out.println(exp);
        }
    }
}
```

Servlet2.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Servlet2 extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter pwriter = response.getWriter();
```

```
//Reading cookies
Cookie c[]=request.getCookies();
//Displaying User name value from cookie
pwriter.print("Name: "+c[1].getValue());
//Displaying user password value from cookie
pwriter.print("Password: "+c[2].getValue());

pwriter.close();
}catch(Exception exp){
    System.out.println(exp);
}
}
}
```

Output

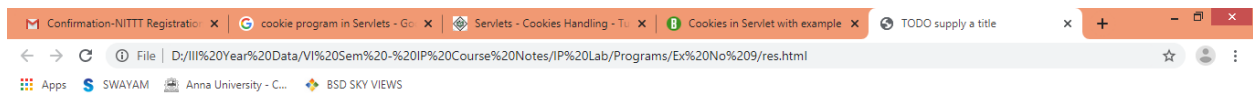


Understanding Cookies. Storing User Name and Password in Cookies

Login Page

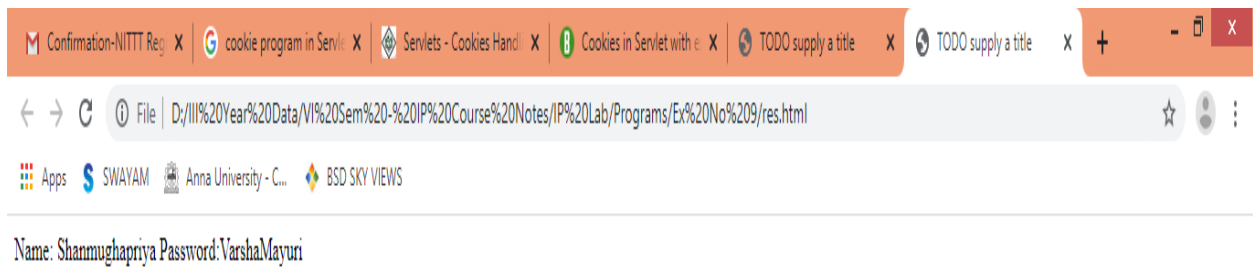
User Name:

Password:



Hello Shanmughapriya Your Password is VarshaMayuri

[View Details](#)



Result

Thus the Servlet program to implement the concept of Cookies has been completed successfully.

XML WITH HTML**AIM:**

To access and display the elements using XML with HTML.

ALGORITHM:

- 1.Start.
- 2.Declare the necessary variables.
- 3.Access the node elements using various in-built methods.
- 4.Print the details.
- 5.Stop.

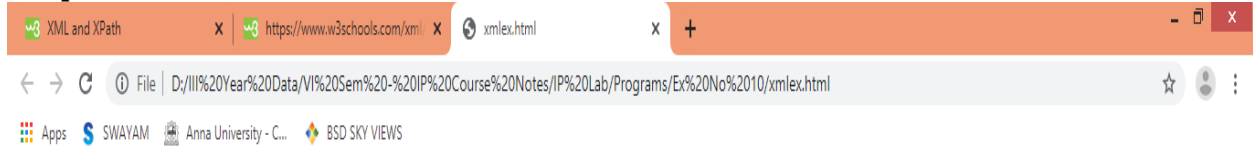
CODING:*HTML File*

```
<html>
<body>
<h1>Reading from a XML FILE</h1>
<div>
<b>To:</b><span id="To"></span><br />
<b>From:</b><span id="From"></span><br />
<b>Message:</b><span id="Message"></span>
</div>
<script>
var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "note.xml", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;
document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

Note.xml

```
<note>
<to>HTML File</to>
<from>XML File</from>
<heading>Reading from XML file- Reading using the
XMLHttp</heading>
<body>Reading using the XMLHttp</body>
</note>
```

Output



Reading from a XML FILE

To:HTML File
From: XML File
Message: Reading using the XMLHttp

Result

Thus the Servlet program to access and display the elements using XML with HTML has been completed successfully.

Ex No 11

Date

XML SCHEMA - XSLT / XSL

AIM:

Program using XML-Schema-XSLT/XSL

ALGORITHM:

Step 1: Start the Program

Step 2: Create a root process for food

Step 3: Create a style for XSLT with focus on each item

Step 4: Output the items

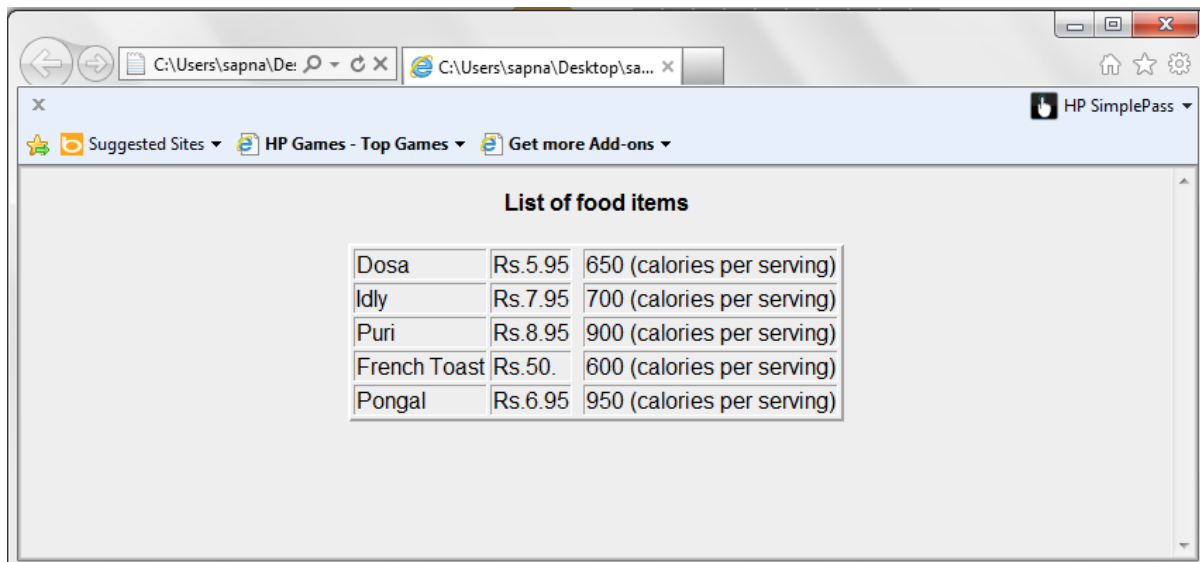
Step 5: Stop

XML CODE

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="eg.xsl"?>
<breakfast_menu>
  <food>
    <name>Dosa</name>
    <price>Rs.5.95</price>
    <calories>650</calories>
  </food>
  <food>
    <name>Idly</name>
    <price>Rs.7.95</price>
    <calories>700</calories>
  </food>
  <food>
    <name>Puri</name>
    <price>Rs.8.95</price>
    <calories>900</calories>
  </food>
  <food>
    <name>French Toast</name>
    <price>Rs.50.</price>
    <calories>600</calories>
  </food>
  <food>
    <name>Pongal</name>
    <price>Rs.6.95</price>
    <calories>950</calories>
  </food>
</breakfast_menu>
```

XSLT CODE

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<html xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<center>
<h4>List of food items</h4>
<table border="2 px">
<xsl:for-each select="breakfast_menu/food">
  <tr>
<td><xsl:value-of select="name"/></td>
<td><xsl:value-of select="price"/></td>
<td><xsl:value-of select="description"/></td>
<td><xsl:value-of select="calories"/>(calories per serving)</td>
  </tr>
</xsl:for-each>
</table>
</center>
</body>
</html>
```



RESULT:

Thus Programs using XML-Schema-XSLT/XSL was developed and successfully executed

Ex No 12

DATE

Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document

AIM

To write a program, that takes user id as input and displays the user details by taking the user information from the XML document

ALGORITHM

Step 1: Start the xml program.

Step 2: Create the 10 user information in the employee tag and insert the information about the employee in

<cse> tag.

Step 3: Then save an XML document by Info.xml which contains 10 user information..

Step 3: Next to retrieve the information by an user id use the java script tag by specifying the script language.

Step 4: Use the function readXMLData() in which information of 10 users can be retrieved by using an id node.

Step 5: Then validate idnode, namenode,department node,designation node by using xmlDocumentObject.getElementsByTagName

Step 6: Next save the file as elements.html and open in the new browser window.

Step 7: Enter id number in the search prompt then the corresponding user details is displayed.

Step 8: Stop the program.

Program

```
<HTML>
<HEAD>
<TITLE>Searching for Elements </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function readXMLData()
{
var xmlDocumentObject, idnode, namenode, designationnode, departmentnode;
xmlDocumentObject=new XMLHttpRequest();
xmlDocumentObject.open("GET","UserInfo.xml",false);
xmlDocumentObject.send();
xmlDocumentObject=xmlDocumentObject.responseXML;
idnode = xmlDocumentObject.getElementsByTagName("UserID");
departmentnode = xmlDocumentObject.getElementsByTagName("UserName");
designationnode = xmlDocumentObject.getElementsByTagName("Phone");
```



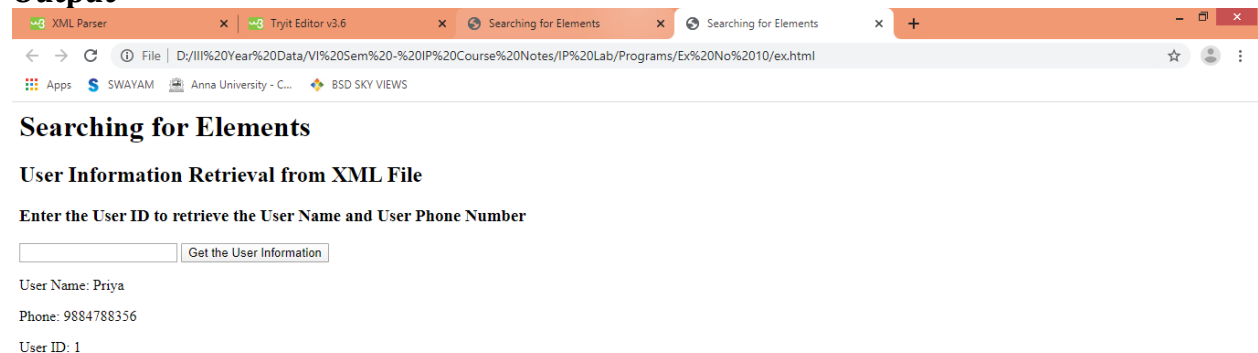
```

namenode = xmlDocumentObject.getElementsByTagName("UserID");
for (i = 0; i < idnode.length; i++)
{
    output=idnode[i].firstChild.nodeValue;
    if (output == document.getElementById("myText").value)
    {
        document.getElementById("displayDIV").innerHTML = idnode[i].firstChild.nodeValue + " "
        + namenode[i].firstChild.nodeValue + " " + departmentnode[i].firstChild.nodeValue + " " +
        designationnode[i].firstChild.nodeValue;
    }
}
}
}
</SCRIPT>
</HEAD>
<BODY>
<H1>Searching for Elements</H1>
<h2>User Information Retrieval from XML File</h2>

<h3>Enter the User ID to retrieve the User Name and User Phone Number</h3>
<input type="text" id="myText" value="">
<input type="BUTTON" VALUE="Get the User Information" ONCLICK="readXMLData()">
<P>
<DIV ID="displayDIV"> </DIV>
</BODY>
</HTML>

```

Output



UserInfo.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates
and open the template in the editor.

```
-->
```

```
<UserInfo>
  <User>
    <UserID>1</UserID>
    <UserName>Priya</UserName>
    <UserPhone>9884788356</UserPhone>
  </User>
  <User>
    <UserID>2</UserID>
    <UserName>Varsha</UserName>
    <UserPhone>9176397027</UserPhone>
  </User>
  <User>
    <UserID>3</UserID>
    <UserName>Karthik</UserName>
    <UserPhone>8825474871</UserPhone>
  </User>
  <User>
    <UserID>4</UserID>
    <UserName>Krithiga</UserName>
    <UserPhone>9884788356</UserPhone>
  </User><User>
    <UserID>5</UserID>
    <UserName>Nagul</UserName>
    <UserPhone>9884788356</UserPhone>
  </User><User>
    <UserID>6</UserID>
    <UserName>Balaji</UserName>
    <UserPhone>9884788356</UserPhone>
  </User>
  <User>
    <UserID>7</UserID>
    <UserName>Jayalakshmi</UserName>
    <UserPhone>9884788356</UserPhone>
  </User>
```

```
<User>
  <UserID>8</UserID>
  <UserName>Priya</UserName>
  <UserPhone>9884788356</UserPhone>
</User>
<User>
  <UserID>9</UserID>
  <UserName>Murugesan</UserName>
  <UserPhone>9884788356</UserPhone>
</User>
<User>
  <UserID>10</UserID>
  <UserName>Kumaravalli</UserName>
  <UserPhone>9884788356</UserPhone>
</User>
</UserInfo>
```

Result:

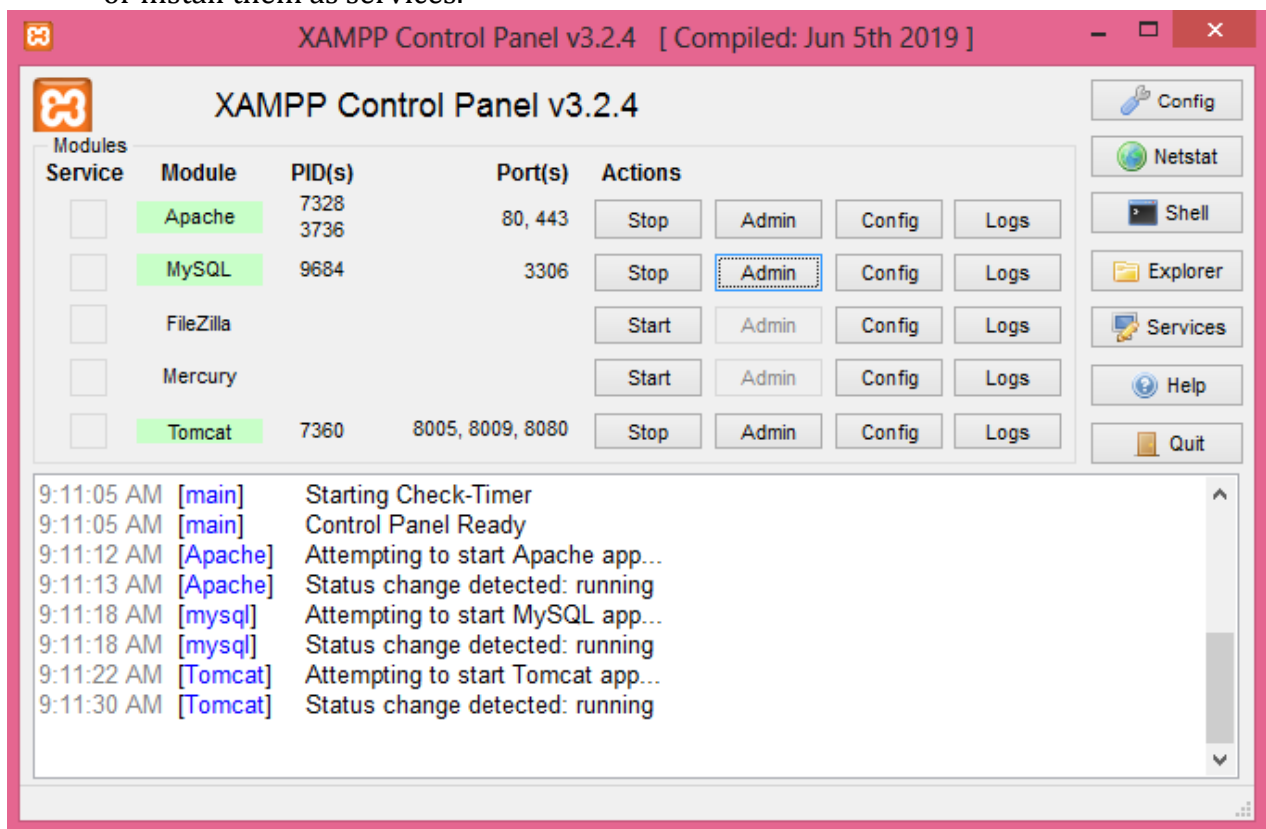
Thus the program which takes user Id as an input and returns the user details by taking the user information from the XML document is done successfully and output was verified.

XAMPP Server Installation to Run PHP and PHP/MYSQL

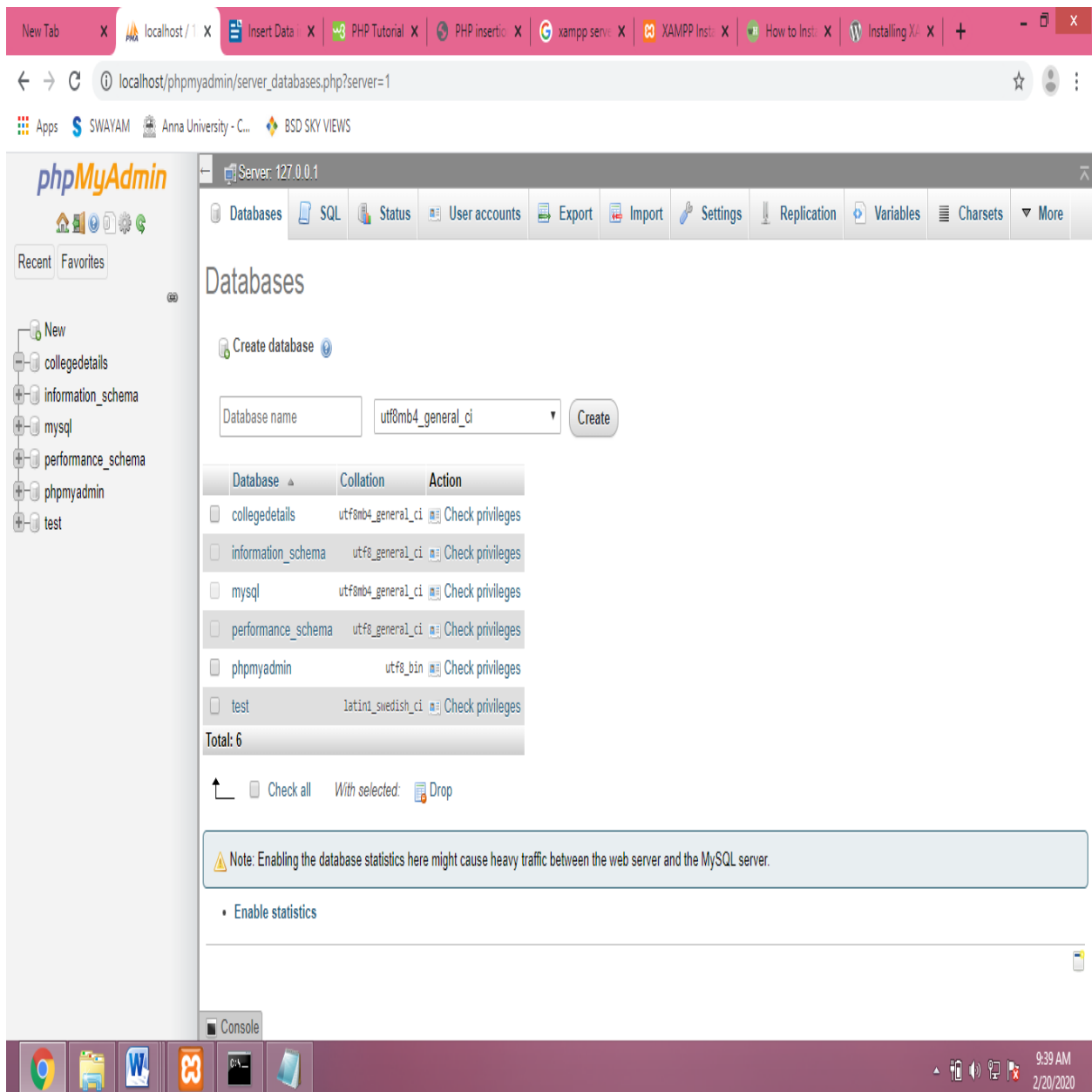
1. Downloading XAMPP #1. Downloading XAMPP
Download the installer file for the latest version of XAMPP, and save the file to your computer.
<https://www.apachefriends.org/index.html>
2. Installing XAMPP #2. Installing XAMPP
3. Next, you need to open the folder where you saved the file, and double-click the installer file.
4. You will be prompted to select the language you wish to use in XAMPP. Click the arrow in the dropdown box, select your language in the list, then click OK to continue the installation process.
5. The Choose Components screen will appear next. This screen will allow you to choose which components you would like to install. To run XAMPP properly, all components checked need to be installed. Click Next to continue.

Starting XAMPP #3. Starting XAMPP

6. The XAMPP Control Panel allows you to manually start and stop Apache and MySQL, or install them as services.



PHP MyAdmin – MYSQL



Place your php files in htdocs folder of C:XAMPP\htdocs and run using `http://localhost/filename.php`

Validate the form using PHP regular expression**AIM:**

To validate the form using PHP regular expression

ALGORITHM:

1. The preg_match() function searches a string for pattern, returning true if the pattern exists, and false otherwise
2. check whether an email address is well-formed is to use PHP's filter_var() function
3. If the URL address syntax is not valid, then store an error message

Program

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
}

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
```

```

// check if e-mail address is well-formed
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $emailErr = "Invalid email format";
}
}

if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid
    if (!preg_match("/\b(?:https?|ftp):\/\/\b|www\b\.)([a-z0-9+&@#\/%?~_!:,;]*[-
a-z0-9+&@#\/%?~_]/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}

if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = test_input($_POST["comment"]);
}

if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = test_input($_POST["gender"]);
}
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

```

```

<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER
["PHP_SELF"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>

```

```

<br><br>
E-mail: <input type="text" name="email">
<span class="error">* <?php echo $emailErr;?></span>
<br><br>
Website: <input type="text" name="website">
<span class="error"><?php echo $websiteErr;?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

```

```

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

```

```

</body>
</html>

```


Output

localhost/ex.php

localhost/ex.php

PHP Form Validation Example

* required field - To be Filled Mandatory

Name: * Name is required

E-mail: * Email is required

Website:

Comment:

Gender: ☐ Female ☐ Male ☐ Other * Gender is required

Submit

Your Input:

localhost/ex.php

localhost/ex.php

PHP Form Validation Example

* required field - To be Filled Mandatory

Name: * SPriyaVarsha

E-mail: * Invalid email format 1768

Website:

Comment:

Gender: ☐ Female ☐ Male ☐ Other * Gender is required

Submit

Your Input:

SPriyaVarsha
1768

localhost/ex.php

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male ☐ Other*

Your Input:

SPriyaVarsha
Shammughapriya.m@gmail.com
www.goog.e.comn
Hello VEC
female

Result

Thus, the program to validate form using PHP was executed and the output was verified successfully.

EX NO 13b

DATE

PHP Program that stores a form data into database

AIM

To write a PHP Program that stores a form data into database

ALGORITHM

1. Create table in MYSQL.
2. Create form in html call php during form submission
3. Create a PHP file to insert data into database.

Create a Database using MYSQL-PHP Admin

Table Creation using MYSQL-PHP Admin

The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL: localhost/phpmyadmin/db_structure.php?server=1&db=collegedetails. The interface is for creating a new table named 'StudentDetail' in the 'collegedetails' database. The table structure is being defined with four columns, all of type 'INT'. The 'Table comments' field is empty, and the 'Storage Engine' is set to 'InnoDB'. The 'PARTITION definition' section is also visible but empty. The left sidebar shows the database structure with 'collegedetails' selected. The bottom status bar shows the time as 9:43 AM on 2/20/2020.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
Student_Name	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---

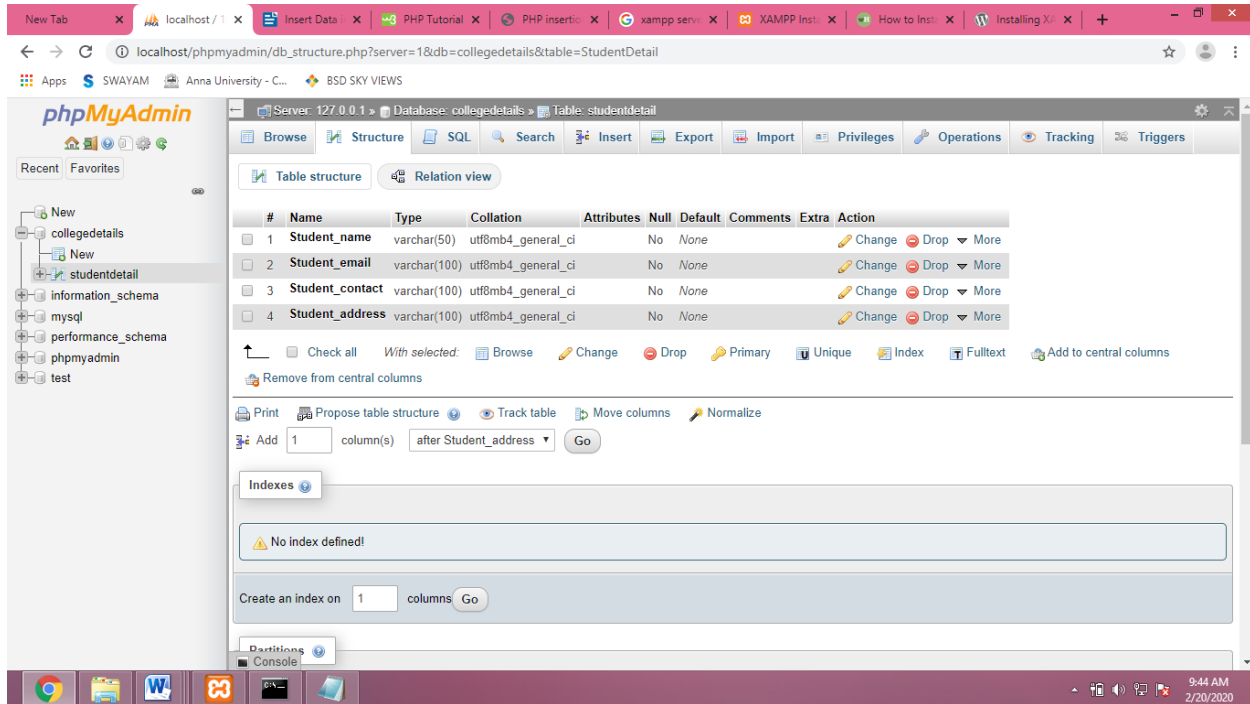
Structure

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Table Created



MY-SQL Code

```
CREATE DATABASE IF NOT EXISTS colleges;
CREATE TABLE students(
student_name varchar(255) NOT NULL,
student_email varchar(255) NOT NULL,
student_contact varchar(255) NOT NULL,
student_address varchar(255) NOT NULL
)
```

HTML File

```
<!DOCTYPE html>
<html>
<head>
<title>PHP insertion</title>
</head>
<body style="background-color:#a9a9a9;">
<center>
<h1>Insert Data In Database Using PHP.</h1>
<form action="insert.php" method="post">
<h2>Basic Details Form</h2>
```

```

<table>
<tr><label>Name:</label></tr>
<tr><input class="input" name="name" type="text" value=""></tr><br><br>
<tr><label>Email:</label></tr>
<tr><input class="input" name="email" type="text" value=""></tr><br><br>
<tr><label>Contact:</label></tr>
<tr><input class="input" name="contact" type="text" value=""></tr><br><br>
<tr><label>Address:</label></tr>
<tr><textarea cols="25" name="address" rows="5"></textarea><br></tr><br><br>
<tr>
<input class="submit" name="submit" type="submit" value="Insert into DataBase">
</center>
</tr>
</table>
</form>

</body>
</html>

```

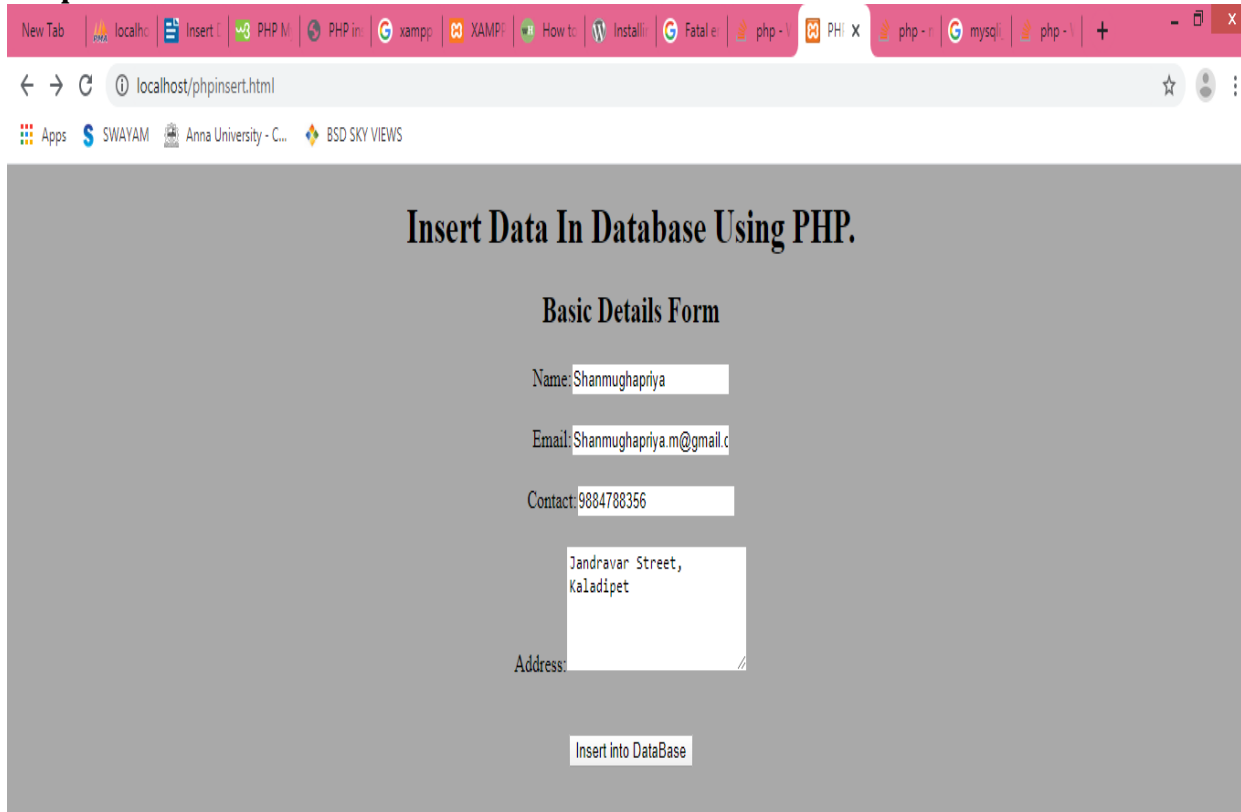
Insert.php

```

<?php
$connection = mysqli_connect("localhost", "root", "");
if ($connection ) {
    die("Connection failed: " . mysqli_connect_error());
}
$db = mysqli_select_db($connection,"collegedetails");
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $email = $_POST['email'];
    $contact = $_POST['contact'];
    $address = $_POST['address'];
    if($name != "" || $email != ""){
        $query = "insert into studentdetails values ('$name', '$email', '$contact', '$address')";
        mysqli_query($db,$query);
        echo "<br/><br/><span>Data Inserted successfully....!!</span>";
    }
    else{
        echo "<p>Insertion Failed <br/> Some Fields are Blank.....!!</p>";
    }
}
mysqli_close($connection); // Closing Connection with Server
?>

```

Output



New Tab | localhost | Insert | PHP M | PHP in | xampp | XAMPP | How to | Install | Fatal e | php - | PH | X | php - | mysql | php - | + | - | X

localhost/phpinsert.html

Apps | SWAYAM | Anna University - C... | BSD SKY VIEWS

Insert Data In Database Using PHP.

Basic Details Form

Name: Shanmughapriya

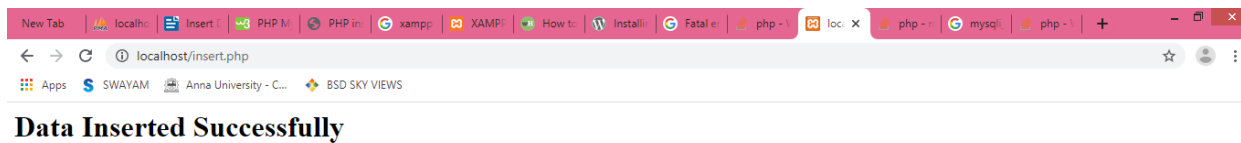
Email: Shanmughapriya.m@gmail.com

Contact: 9884788356

Jandrar Street,
Kaladipet

Address:

Insert into DataBase



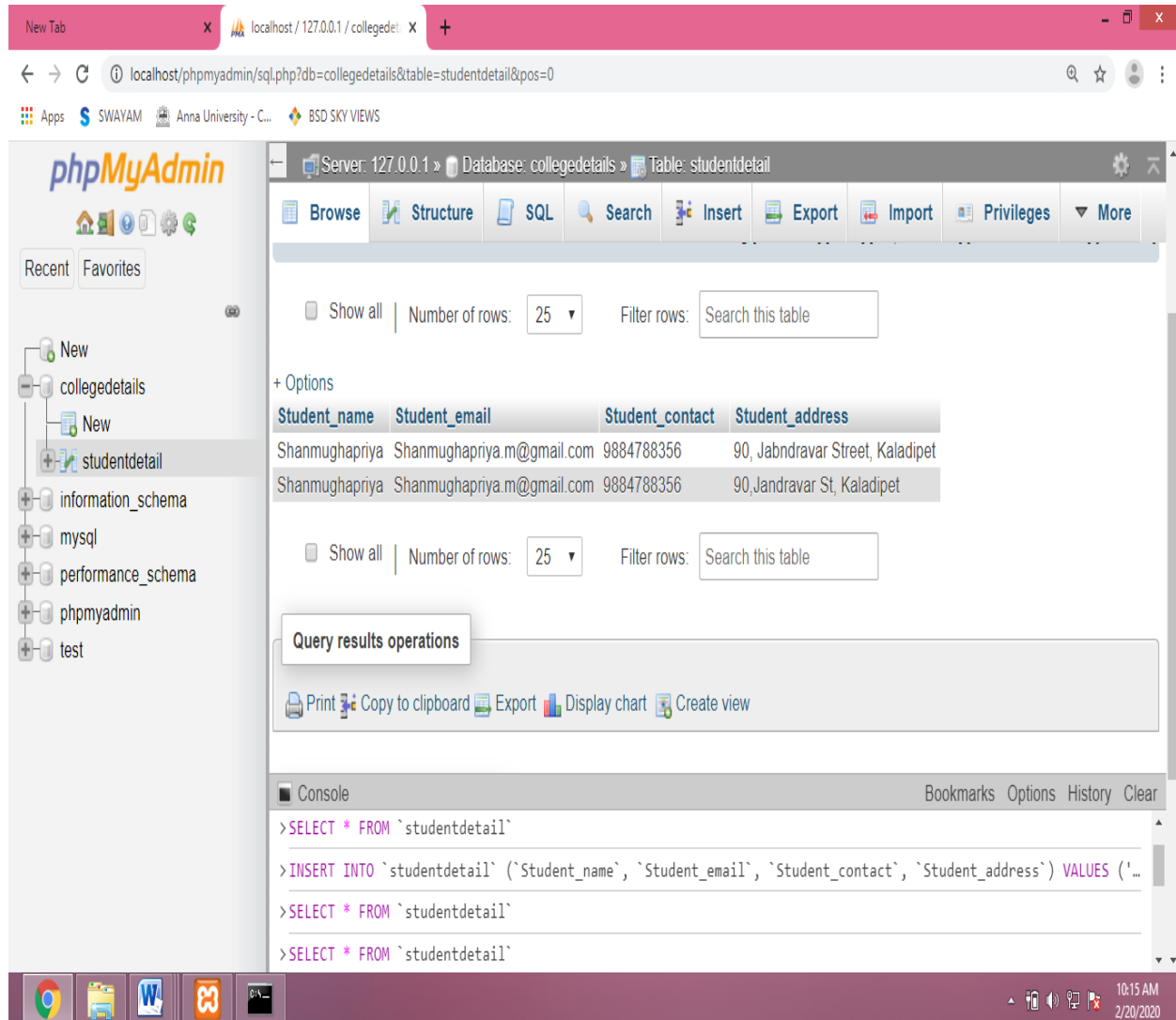
New Tab | localhost | Insert | PHP M | PHP in | xampp | XAMPP | How to | Install | Fatal e | php - | loc: X | php - | mysql | php - | + | - | X

localhost/insert.php

Apps | SWAYAM | Anna University - C... | BSD SKY VIEWS

Data Inserted Successfully

PHP MyAdmin – Inserted Data View



The screenshot displays the PHP MyAdmin web interface. The browser address bar shows the URL: `localhost/phpmyadmin/sql.php?db=collegedetails&table=studentdetail&pos=0`. The interface shows the 'collegedetails' database selected, with the 'studentdetail' table highlighted in the left sidebar. The table structure is visible, showing columns: `Student_name`, `Student_email`, `Student_contact`, and `Student_address`. The table contains two rows of data:

Student_name	Student_email	Student_contact	Student_address
Shanmughapriya	Shanmughapriya.m@gmail.com	9884788356	90, Jabndravar Street, Kaladipet
Shanmughapriya	Shanmughapriya.m@gmail.com	9884788356	90, Jandravar St, Kaladipet

The console at the bottom shows the following SQL commands:

```
>SELECT * FROM `studentdetail`  
>INSERT INTO `studentdetail` (`Student_name`, `Student_email`, `Student_contact`, `Student_address`) VALUES ('...'  
>SELECT * FROM `studentdetail`  
>SELECT * FROM `studentdetail`
```

RESULT

Thus, the program to insert data from form using PHP to a database was executed and the output was verified successfully.

Ex No 14

DATE

Write a web service for finding what people think by asking 500 people's opinion for any consumer product.

AIM

To write a web service for finding what people think by asking 500 people's opinion for any consumer product.

ALGORITHM:

Step 1. Start

Step 2. Create a new project under Java Web->Web Application.

Step 3. Right click on the source package and create new Web Service.

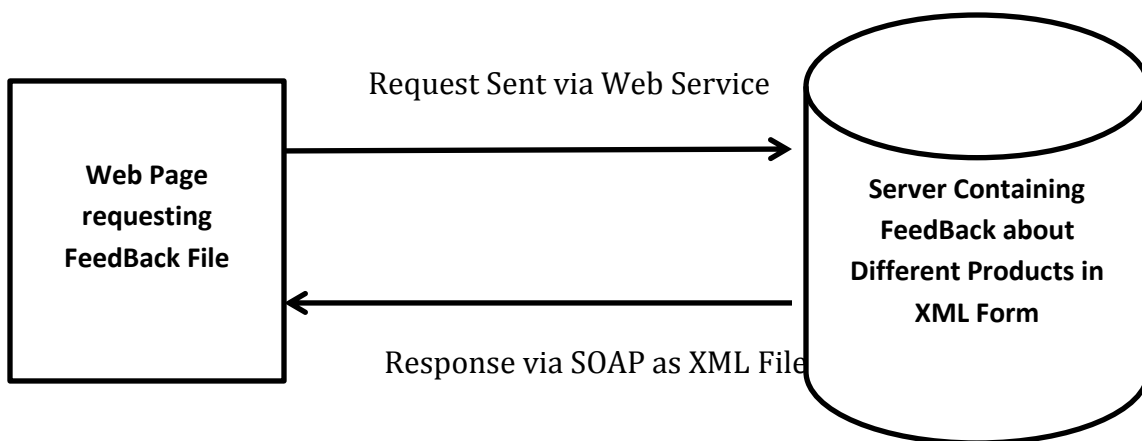
Step 4. Define necessary with its parameters and operations.

Step 5. Right click on the project name and select Clean and Build. After completion, right click again on the project name and select deploy.

Step 6. Right click on Web Services and select Test Web Service to run the program.

Step 7. End

WebService Architecture



CustomerFeedBack.java

```
import java.io.Serializable;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;
@XmlRootElement(name = "user")

public class CustomerFeedBack implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private String product;
    private String feedback;
    public User(){

    }

    public User(int id, String UserName, String Product, String FeedBack ){
        this.id = id;
        this.name = UserName;
        this.product = Product;
        this.feedback=FeedBack;
    }
    public int getId() {
        return id;
    }
    @XmlElement
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    @XmlElement
    public void setName(String name) {
        this.name = name;
    }
    public String getProduct() {
        return product;
    }
    @XmlElement
    public void setProduct(String product) {
        this.product = product;
    }
    public String getFB() {
        return feedback;
    }
    @XmlElement
```

```

    public void setFB(String fb) {
        this.feedback = fb;
    }
}

```

GetFeedBack.java

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.List;

public class GetFeedBack {
    public List< CustomerFeedBack > getAllUsers(){

        List< CustomerFeedBack > userFBList = null;
        try {
            File file = new File("CustomerFeedBack.dat");
            if (!file.exists()) {
                CustomerFeedBack fb = new CustomerFeedBack (1, "Priya","LG-TV","Good");
                userFBList = new ArrayList< CustomerFeedBack >();
                userFBList.add(fb);
                saveFBList(userFBList);
            }
            else{
                FileInputStream fis = new FileInputStream(file);
                ObjectInputStream ois = new ObjectInputStream(fis);
                userFBList = (List< CustomerFeedBack >) ois.readObject();
                ois.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return userFBList;
    }
    private void saveFBList(List< CustomerFeedBack > userFBList){
        try {
            File file = new File("CustomerFeedBack.dat");
            FileOutputStream fos;
            fos = new FileOutputStream(file);

```

```

        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(userFBList);
        oos.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

UserFBService.java

```

import java.util.List;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
@Path("/UserFBService")

public class UserFBService {
    GetFeedBack Get_FB = new GetFeedBack ();
    @GET
    @Path("/users")
    @Produces(MediaType.APPLICATION_XML)
    public List< CustomerFeedBack > getUsers(){
        return GetFeedBack.getAllUsers();
    }
}

```

FEEDBACK Using PHP – Storing Form Data as XML File which can be sent/ read from a Web Service

Feedback.html

```
<html>
<head>
<title>
Customer Feedback Form
</title>
</head>
<body >
<center>
<br><br><br><br>
<h1> Customer Feedback Form</h1>
<form action="writexmlfile.php" method="POST">
<table>
<tr>User ID : <input type="text" id="userid"></tr><br><br>
<tr>User Name : <input type="text" id="username"></tr><br><br>
<tr>Product Name : <input type="text" id="product"></tr><br><br>
<tr>FeedBack : <input type="text" id="fb"></tr><br><br><br>
<input type="submit" value="Save to File">
</form>
</table>
</center>

</body>

</html>
```

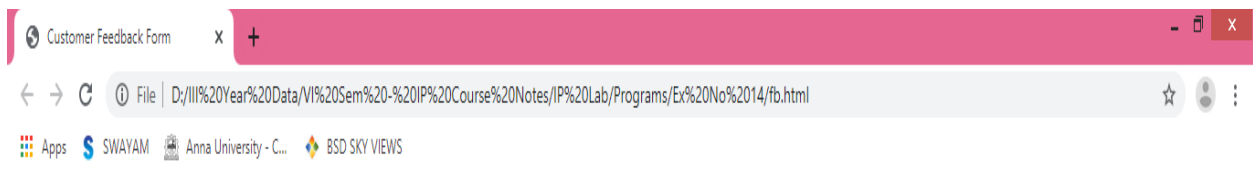
Writexmlfile.php

```
<?php>
$a,$b,$b,$d;
$a=$_POST["userid"];
$b=$_POST["username"];
$c=$_POST["product"];
$d=$_POST["fb"];
$xml = new DomDocument();
$xml->load("form.xml");
$xml->formatOutput = true;
$global = $xml->documentElement;
$global = $xml->firstChild;
$item0=$xml->createElement("User");
$item1 = $xml->createElement("UserID");
$item2 = $xml->createElement("UserName");
$item3 = $xml->createElement("Product");
$item4 = $xml->createElement("FeedBack");
```

```
$item0->appendChild($item1);  
$item0->appendChild($item2);  
$item0->appendChild($item3);  
$item0->appendChild($item4);  
$item1->nodeValue=$a;  
$item2->nodeValue=$b;  
$item3->nodeValue=$c;  
$item4->nodeValue=$d;  
$xml->save("form.xml");
```

?>

Output



Customer Feedback Form

User ID :

User Name :

Product Name :

FeedBack :

```
<?xml version='1.0'>
<User-FB>
  <User>
    <UserID>1</UserID>
    <UserName>Shanmughapriya</UserName>
    <Product>LG</Product>
    <FeedBack>Good</FeedBack>
  </User>
  <User>
    <UserID>1</UserID>
    <UserName>Shanmughapriya</UserName>
    <Product>LG</Product>
    <FeedBack>Good</FeedBack>
  </User>
  <User>
    <UserID>1</UserID>
    <UserName>Shanmughapriya</UserName>
    <Product>LG</Product>
    <FeedBack>Good</FeedBack>
  </User>
  <User>
    <UserID>1</UserID>
    <UserName>Shanmughapriya</UserName>
    <Product>LG</Product>
    <FeedBack>Good</FeedBack>
  </User>
</User-FB>
```

Result

The program to write a web service for finding what people think by asking 500 people's opinion for any consumer product has been completed successfully.

Content beyond Syllabus

Machine Learning with Java Script – Program to Auto Change Text Color with respect to background color contrast in web sites

Brain.js – Machine Learning Library

Reference:

How to make a simple Machine Learning Website from scratch

<https://hackernoon.com/how-to-make-a-simple-machine-learning-website-from-scratch-1ae4756c8b04>

Code Snippet – Outlines the use of brain.js framework

```
// create new neural network object
const network = new brain.NeuralNetwork();

// training data
network.train([
  { input: { r: 0.62, g: 0.72, b: 0.88 }, output: { light: 1 } },
  { input: { r: 0.1, g: 0.84, b: 0.72 }, output: { light: 1 } },
  { input: { r: 0.33, g: 0.24, b: 0.29 }, output: { dark: 1 } },
  { input: { r: 0.74, g: 0.78, b: 0.86 }, output: { light: 1 } },
  { input: { r: 0.31, g: 0.35, b: 0.41 }, output: { dark: 1 } },
  { input: { r: 1, g: 0.42, b: 0.52 }, output: { dark: 1 } },
  { input: { r: 0, g: 0, b: 1 }, output: { dark: 1 } },
  { input: { r: 0.8, g: 0.44, b: 1 }, output: { dark: 1 } },
  { input: { r: 0, g: 0.44, b: 1 }, output: { dark: 1 } },
  { input: { r: 0.3, g: 0.6, b: 1 }, output: { dark: 1 } },
  { input: { r: 0.1, g: 0.6, b: 0 }, output: { dark: 1 } }
]);

// grab inputs in .rgbValues div
var input = document.querySelectorAll(".rgbValues input");
// grab inputs in .colorPicker div
var sliderInput = document.querySelectorAll('.colorPicker input');

// rgb slider
for(var i = 0; i < sliderInput.length; i++){
  sliderInput[i].addEventListener("input", function(){
    var redSlider = document.getElementById("redSlider").value;
```

```

        var greenSlider = document.getElementById("greenSlider").value;
        var blueSlider = document.getElementById("blueSlider").value;

        document.getElementById("red").value = redSlider;
        document.getElementById("green").value = greenSlider;
        document.getElementById("blue").value = blueSlider;

        changeBackgroundColor(redSlider, greenSlider, blueSlider);

        var networkObject = createNetworkObject(redSlider, greenSlider, blueSlider);
        var MLresult = network.run(networkObject);

        MLresult = networkLabel(MLresult);
        console.log(MLresult);

        changeElementsColor(MLresult, input, sliderInput);
    });
}

// rgb value input (the non slider one)
for(var i = 0; i < input.length; i++){
    input[i].addEventListener("keyup", function(e){
        // right arrow key to next tab
        if(e.keyCode == 39){
            $(this).next('input, select').focus();
        }
        // left arrow key to previous tab
        if(e.keyCode == 37){
            $(this).prev('input, select').focus();
        }
        // max value input 255
        if(e.target.value > 255){
            e.target.value = 255;
        }
    });

    var container = document.getElementById("container");
    var red = document.getElementById("red").value;
    var green = document.getElementById("green").value;
    var blue = document.getElementById("blue").value;

    // change slider values to correspond to rgb value input(s)

```



```

        document.getElementById('redSlider').value = red;
        document.getElementById('greenSlider').value = green;
        document.getElementById('blueSlider').value = blue;

        changeBackgroundColor(red, green, blue);

        // machine learning
        var networkObject = createNetworkObject(red,green,blue);
        var MLresult = network.run(networkObject);

        MLresult = networkLabel(MLresult);
        console.log(MLresult);

        changeElementsColor(MLresult, input, sliderInput);
    });

    // when input changes from up down arrows keys to change value
    input[i].addEventListener("change", function(){
        var container = document.getElementById("container");
        var red = document.getElementById("red").value;
        var green = document.getElementById("green").value;
        var blue = document.getElementById("blue").value;

        changeBackgroundColor(red, green, blue);

        // machine learning
        var networkObject = createNetworkObject(red,green,blue);
        var MLresult = network.run(networkObject);

        MLresult = networkLabel(MLresult);
        console.log(MLresult);

        var MLresult = network.run(networkObject);

        // black text for 'light' colours
        changeElementsColor(MLresult, input, sliderInput);
    });
}

```