**EXPERIMENT-1**

**AIM:** Design static web sites with html tags by taking different examples.

**DESCRIPTION:** Here,we are using different html tags to build a static web page.There are some tags which must be there to run a html program.They are : head

body

html

The tags in **HTML** contains opening and closing tags.like <head></head>,< body></body>, <html></html>.Closing tag contains a special character ‘/’ before the tagname.

**<!DOCTYPE html>**  is the first line in an html program, It is an information to the browser about what document type to expect.

Tags used in this program are h1-h6 tags for headings,<u></u> for underlines, <img> for inserting images, <table></table> for creating table, and <th>,<tr> and <td> for table contents.

**PROGRAM:**

<html>

<head>

</head>

<body>

<center>

<h1>CV/RESUME</h1>

</center>

<br><br>

<img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images folder\image.jpeg" width="120" height="100" align="right"/>

<h3>To work in an organisation which provides me with ample opportunities to enhance my skills and knowledge along with contributing to the growth of the organisation.</h3>

<h3>Create value for my prospective employer through my dedicated and diligent efforts.</h3>

<h4 align="right">NAME : Snehita<br>ROLL NO : 528 </h4>

<hr>

<br>

<p><i>This is Snehita.Studying B.Tech in PVP Siddhartha Institute Of Technology...</i></p>

<h3><u>Cetifications</u></h3>

<p><a href=" C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images folder\certificate\_img.jpeg ">WEB APPS Certification</a></p>

<h4><u>Hobbies</u></h4>

<h3><font face="arial" color="red" size="3">Listening to music<br>Singing<br>Watching Music<br>Learning new things</font></h3>

<div id="myDIV"><img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 2\image.jpg" width="300" align="right" height="200px"></div>

<h3><u>MARKS</u></h3>

<table width="80" border="1">

<tr>

<th>SSC</th>

<th>Inter</th>

<th>BTech</th>

</tr>

<tr>

<td>9.7</td>

<td>9.6</td>

<td>9.09</td>

</tr>

</table>

</body>

</html>

**WEB APPS Certification file:**

<html>

<body>

<h1>WEB APPS Certification</h1>

<h2>Certificate Of Completion</h2>

<img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images

folder\web-development-Certificate.jpg" width=1325" height="500" />

</body>

</html**>**

**OUTPUT:** Graphical user interface, text, application, email

Description automatically generated

* When the WEB APPS Certification is clicked,the certificate page will be displayed.

Graphical user interface, text, application

Description automatically generated

**EXPERIMENT-2**

**AIM:** Design web pages using different types of CSS.

**DESCRIPTION:** Applying different CSS for a Static web page.There are 3 types of CSS.

They are : Inline CSS, Internal CSS, External CSS.

Inline CSS - Which is written inside the elements opening tags of body section.

Internal CSS - Which is written inside the head section.

External CSS – Which is written in a separate file and combine with a link called link tag in head section.

**PROGRAM:**

**1)**

<html>

<head>

<link rel="stylesheet" href="externalcssapp.css">

<style>

body{

background-image:url("bgimage.jpg");

background-size:cover;

}

.container{

text-align: right;

}

.grid-container {

display: grid;

grid-template-columns: 80px 80px 80px;

padding: 10px;

}

.grid-item {

background-color: rgba(255, 255, 255, 0.8);

border: 1px solid rgba(0, 0, 0,0.8);

padding: 20px;

font-size:20px;

text-align: center;

}

#myDIV {

width: 300px;

height: 200px;

border: 1px solid black;

animation: mymove 5s infinite;

transform: scale();

}

@keyframes mymove {

50% {border-radius: 50px;}

}

</style>

</head>

<body>

<div class="container">

<h1 style="text-align:center; color:green;">CV/RESUME</h1>

<br><br>

<img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images folder\image.jpeg" width="120" height="100" align="right"/>

<h3 style="text-align:left; font-family:cursive;">To work in an organisation which provides me with ample opportunities to enhance my skills and knowledge along with contributing to the growth of the organisation.</h3>

<h3 style="text-align:left; font-family:cursive;">Create value for my prospective employer through my dedicated and diligent efforts.</h3>

<h4 >NAME : Snehita<br>ROLL NO : 528 </h4>

<hr>

</div>

<p style="font-size:20px"><i>This is Snehita.Studying B.Tech in PVP Siddhartha Institute Of Technology...</i></p>

<div>

<h3><u>Cetifications</u></h3>

<p><a href=" C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images folder\certificate\_img.jpeg ">WEB APPS Certification</a></p>

<h4 style="font-size:20px;"><u>Hobbies</u></h4>

<h3 style="font-weight: normal; font-family:arial;">Listening to music<br>Singing<br>Watching Music<br>Learning new things</h3>

</div>

<div id="myDIV"><img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 2\image.jpg" width="300" align="right" height="200px"></div>

<h3><u>MARKS</u></h3>

<div class="grid-container">

<div class="grid-item">SSC</div>

<div class="grid-item">Inter</div>

<div class="grid-item">B.Tech</div>

<div class="grid-item">9.7</div>

<div class="grid-item">9.6</div>

<div class="grid-item">9.09</div>

</div>

<h5>HEY!This is External CSS</h5>

<h2>External CSS is applied</h2>

</body>

</html>

**EXTERNAL FILE(externalcssapp.css) :**

h5{

font-family:cursive;

font-color:red;

}

h2{

font-family:italic;

font-color:blue;

font-size:25px;

}

**WEB APPS Certification file:**

<html>

<body>

<h1>WEB APPS Certification</h1>

<h2>Certificate Of Completion</h2>

<img src="C:\Users\snehi\OneDrive\Desktop\WAD\EXP 1\images

folder\web-development-Certificate.jpg" width=1325" height="500" />

</body>

</html**>**

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

* When the WEB APPS Certification is clicked,the certificate page will be displayed.

Graphical user interface, text, application

Description automatically generated

**EXPERIMENT-3**

**AIM:** Apply Client side validations using JavaScript.

**DESCRIPTION:**

Tags used are:

div

label

input

form

script

This Java Script is written inside the html program and that can be written within script tags of html program.

**PROGRAM:**

**JS Validation file:**

<!DOCTYPE html>

<html>

<head>

<style>

html{

height:100%;

}

body{

text-align:center;

}

.Login-box{

padding:200px 200px 200px 200px;

border-sizing:border-box;

border:2px solid red;

border-radius:20px;

text-align:center;

}

.login-box img{

width:100px;

height:100px;

}

.login-box h3{

margin:0px 0px 30px;

padding :0px;

}

.login-box username-box input{

width:100px;

padding:10px 0;

font-size:18px;

margin-buttom:30px;

}

.login-box username-box label{

text-align:left;

padding:0 0 10px 0;

font-size:16px;

transition: .5s;

}

</style>

</head>

<script type="text/javascript">

function validate() {

let username=usr\_form.f\_u\_name.value;

let re\_enter=usr\_form.p\_name.value;

//let result = username.localeCompare(re\_enter);

if(username=="usqt94" && re\_enter=="usqt94")

{

return true

}

else{

console.log(re\_enter);

alert("Doesnt match")

return false

}

}

</script>

<body>

<div class="Login-box">

<div class="row">

<img src="C:\Users\snehi\OneDrive\Desktop\WAD\Exam1\images folder exam\login page boy image.jpg" width="100px" height="100px"/>

</div>

<h3>Login Account</h3>

<form action="successpage.html" name="usr\_form" onsubmit="return validate()">

<div class="username-box">

<label>Username</label><br>

<input type="text" name="f\_u\_name"></input>

<br>

<label>Re-enter Username</label><br>

<input type="text" name="p\_name"></input>

<br>

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

</div>

</form>

</div>

</body>

</html>

**Successpage file:**

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

<h1>Success</h1>

</body>

</html>

**OUTPUT:**

**1)**

Graphical user interface, application, Word

Description automatically generated

**2)**

Graphical user interface, application, Word

Description automatically generated

**3)**

Graphical user interface, text, application, Word

Description automatically generated

**4)**

Graphical user interface, application, Word

Description automatically generated

**5)**

Graphical user interface, Word

Description automatically generated

**EXPERIMENT-4**

**AIM:**  Create an XML file for student/employee/book data and validate against DTD.

**DESCRIPTION:** XML is an Extensible Mark up language, similar to HTML.

* In XML, there will be using User-defined tags.

Validating an XML document: Process of checking a document written in XML to confirm that it is both well formed and also ‘valid’ in that follows a defined structure.

* It displays on the web page, what is written in XML document.

**DTD:**

* DTD stands for Document type Definition.
* It is a way to describe XML Language precisely.
* It checks validity & vocabulary of the XML document structure against grammatical rules of appropriate XML language.
* **Syntax:** <!DOCTYPE element DTP identifier[declarations….]>
* There are **2** types of DTD. They are Internal DTD and External DTD.

**Internal DTD:** It means,if elements are declared within the XML files.

* To specify it as internal DTD, standalone attribute in XML declaration must be set to **“yes”.**

**Syntax:**  <!DOCTYPE root-element [element-declaration]>

**External DTD:** It means, if DTD elements are declared outside the XML file.

* To specify it as external DTD, standalone must be set to **“no”.**

**Syntax:**  <!DOCTYPE root-element SYSTEM “file-name.dtd”>

* Where file name is the file with **.dtd** extension.

**PROGRAM:**

**WITHOUT USING DTD:**  
<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>  
<Student\_info>  
        <pvp>  
          <Name>  
           <FirstName>Snehita</FirstName>  
           <MiddleName> </MiddleName>  
           <LastName>Bharata</LastName>  
          </Name>  
          &#xA;  
          <Study>  
               <year>III</year>  
               <section>S1</section>  
               <department>CSE</department>  
          </Study>  
          &#xA;  
          <Contact\_details>  
               <mob\_no>9876543210</mob\_no>  
               <email>[snehita@gmail.com](mailto:snehita@gmail.com)</email>  
               <address>Kanuru,Vijayawada</address>  
          </Contact\_details>    
      </pvp>  
</Student\_info>

**USING INTERNAL DTD:**

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes" ?>  
<!DOCTYPE Student\_info [  
   <!ELEMENT pvp (Name,Study,Contact\_info)>  
   <!ELEMENT Name (FirstName,MiddleName,LastName)>  
   <!ELEMENT Study (year,section,department)>  
   <!ELEMENT Contact\_details (mob\_no,email,address)>  
   <!ELEMENT FirstName (#PCDATA)>  
   <!ELEMENT MiddleName (#PCDATA)>  
   <!ELEMENT LastName (#PCDATA)>  
   <!ELEMENT year ( #PCDATA)>  
   <!ELEMENT section (#PCDATA)>  
   <!ELEMENT department (#PCDATA)>  
   <!ELEMENT mob\_no (#PCDATA)>  
   <!ELEMENT email (#PCDATA)>  
   <!ELEMENT address (#PCDATA)>  
]>  
  
<Student\_info>  
         <Name>  
           <FirstName>Snehita</FirstName>  
           <MiddleName> </MiddleName>  
           <LastName>Bharata</LastName>  
          </Name>  
          &#xA;  
          <Study>  
               <year>III</year>  
               <section>S1</section>  
               <department>CSE</department>  
          </Study>  
          &#xA;  
          <Contact\_details>  
               <mob\_no>9876543210</mob\_no>  
               <email>[snehita@gmail.com](mailto:snehita@gmail.com)</email>  
               <address>Kanuru,Vijayawada</address>  
          </Contact\_details>    
</Student\_info>

**EXTERNAL DTD:**

<?xml version = "1.0" encoding = "UTF-8" standalone = "no" ?>  
<!DOCTYPE Student\_info SYSTEM "**Student\_info.dtd**">  
<Student\_info>  
         <Name>  
           <FirstName>Snehita</FirstName>  
           <MiddleName> </MiddleName>  
           <LastName>Bharata</LastName>  
          </Name>  
          &#xA;  
          <Study>  
               <year>III</year>  
               <section>S1</section>  
               <department>CSE</department>  
          </Study>  
          &#xA;  
          <Contact\_details>  
               <mob\_no>9876543210</mob\_no>  
               <email>[snehita@gmail.com](mailto:snehita@gmail.com)</email>  
               <address>Kanuru,Vijayawada</address>  
          </Contact\_details>    
</Student\_info>  
  
 **Student\_info.dtd file:**  
   
 <!ELEMENT pvp (Name,Study,Contact\_info)>  
   <!ELEMENT Name (FirstName,MiddleName,LastName)>  
   <!ELEMENT Study (year,section,department)>  
   <!ELEMENT Contact\_details (mob\_no,email,address)>  
   <!ELEMENT FirstName (#PCDATA)>  
   <!ELEMENT MiddleName (#PCDATA)>  
   <!ELEMENT LastName (#PCDATA)>  
   <!ELEMENT year ( #PCDATA)>  
   <!ELEMENT section (#PCDATA)>  
   <!ELEMENT department (#PCDATA)>  
   <!ELEMENT mob\_no (#PCDATA)>  
   <!ELEMENT email (#PCDATA)>  
   <!ELEMENT address (#PCDATA)>

**OUTPUT:**

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

**EXPERIMENT-5**

**A)**

**AIM:** Develop different JDBC applications to interact with database.

**DESCRIPTION:** JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE(Java Standard Edition).

**Using JDBC to connect to a database:**

* Install or locate the database you want to access.
* Include the JDBC library.
* Ensure the JDBC driver you need is on your classpath.
* Use the JDBC library to obtain a connection to the database.
* Use the connection to issue SQL commands.
* Close the connection when you're finished.

**Steps to connect data base with driver:**

1)Install mysql installer.

Graphical user interface, text, application, email

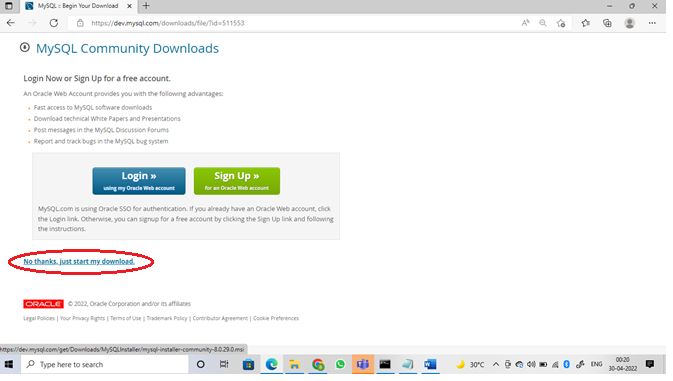
Description automatically generated

2)Select Platform Independent for selecting OS.

Graphical user interface, text, application, Word

Description automatically generated

3)Download 32-bit installer for windows, and start downloading.

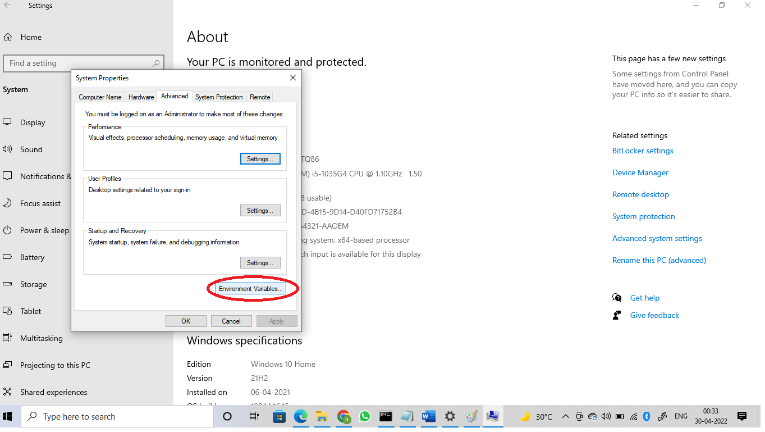


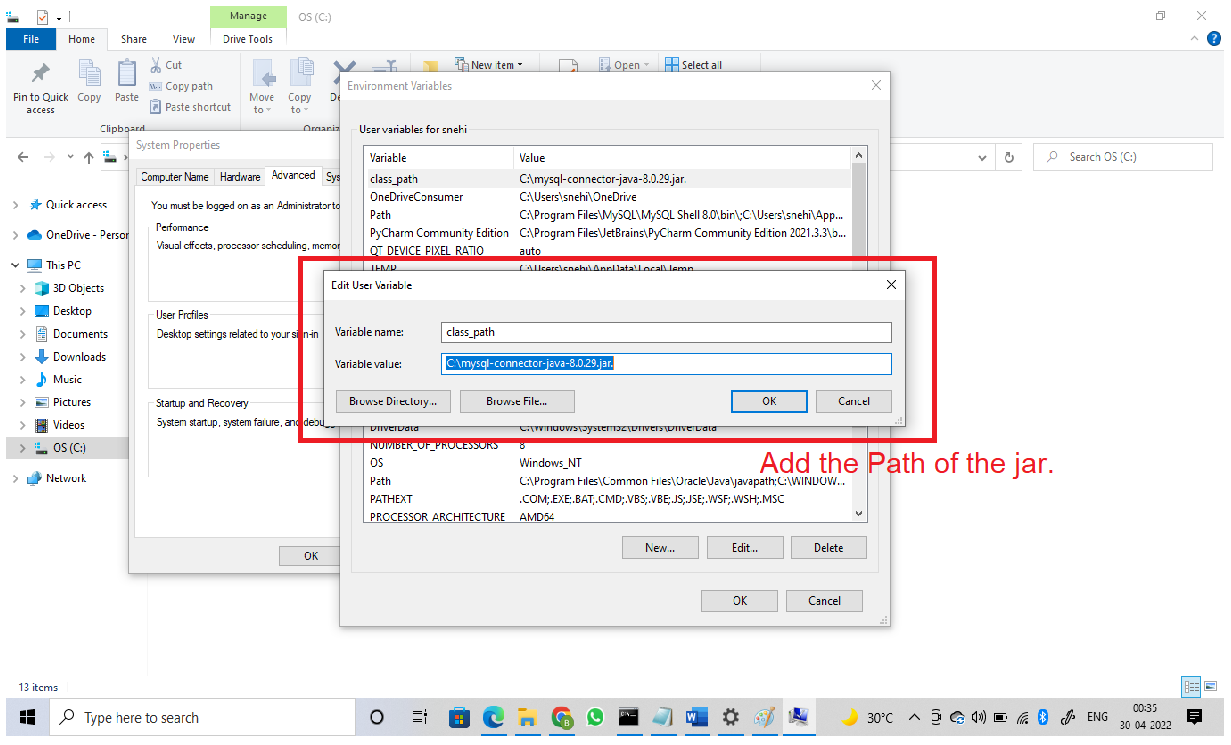
4)Extract/Unzip the downloaded jar Zip file.

5) Add the jar file in program files->java->jdk->lib

6)Go to Properties of the jar file, go to advanced settings and set environment variables by adding or editing the new class\_path variable referencing the location of the jar file.







7)Write java Program in Editor(Notepad/Notepad++,etc..) and in the program we should give the root name and password of our database.

**PROGRAM:**

import java.util.\*;

import java.sql.\*;

import java.sql.Connection;

import java.sql.DriverManager;

public class JDBCExample {

public static void main(String[] args) throws Exception {

System.out.println("hiiiiiiii");

Class.forName("com.mysql.cj.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306";

String username = "root";

String password ="snehita2002";

Connection con = DriverManager.getConnection(url, username, password);

if (con != null) {

System.out.println("Database Connected successfully");

} else {

System.out.println("Database Connection failed");

}

}

}

8)Save the class name and file name as same.

9)Execute java Program using Java Commands in Command Prompt.

10)Finally, it shows whether Database is Connected or not.

**OUTPUT:**

**Text

Description automatically generated**

**B)**

**AIM:** Write a Java Program which retrieves data from the database.

**PROGRAM:**

import java.util.\*;

import java.sql.\*;

class prepstmt1{

public static void main(String args[]) {

try {

int rows=0;

Class.forName("com.mysql.cj.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","snehita2002");

String sql = "insert into details values(?,?,?,?)";

PreparedStatement ps = c.prepareStatement(sql);

ps.setString(1,"Keerthika");

ps.setString(2,"19501a0535");

ps.setString(3,"CSE");

ps.setString(4,"1");

rows = ps.executeUpdate();

if(rows>0) {

System.out.println("Successfully inserted");

c.close();

}

}

catch (Exception e) {

System.out.println(e);

}

}

}

**OUTPUT:**

Text

Description automatically generated

A picture containing text

Description automatically generated

**EXPERIMENT-6**

**A)**

**AIM:** Create different web applications using servlets.

**DESCRIPTION:**

**Servlet -** Servlet is a technology which is used to create a web application. Servlet is an API that provides many interfaces and classes including documentation. Servlet is an interface that must be implemented for creating any Servlet. Servlet is a class that extends thecapabilities of the servers and responds to the incoming requests. It can respond to any

requests. Servlet is a web component that is deployed on the server to create a dynamic web

page.

Classes in javax.servlet.http package

There are many classes in javax.servlet.http package. They are as follows:

1. HttpServlet

2. Cookie

3. HttpServletRequestWrapper

4. HttpServletResponseWrapper

5. HttpSessionEvent

Interfaces in javax.servlet.http package

There are many interfaces in javax.servlet.http package. They are as follows:

1. HttpServletRequest

2. HttpServletResponse

3. HttpSession

Methods of HttpServlet class

There are many methods in HttpServlet class. They are as follows:

1. **public void service(ServletRequest req,ServletResponse res)** dispatches the request

to the protected service method by converting the request and response object into http type.

2. **protected void service(HttpServletRequest req,HttpServletResponse res)** receives the request from the service method, and dispatches the request to the doXXX() method depending on the incoming http request type.

3. **protected void doGet(HttpServletRequest req, HttpServletResponse res)** handles

the GET request. It is invoked by the web container.

4. **protected void doPost(HttpServletRequest req, HttpServletResponse res)** handles

the POST request. It is invoked by the web container.

5. **protected void doDelete(HttpServletRequest req, HttpServletResponse res)** handles the DELETE request. It is invoked by the web container.

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<form action="Multiple.java" method="post">

Enter number1:<input type="text" name="num1"><br><br>

Enter number2:<input type="text" name="num2"><br><br>

<input type="submit">

<input type="reset">

</form>

</body>

</html>

**Multiple.java**

import javax.servlet.http.\*;

import java.io.\*;

public class Multiple extends HttpServlet {

public void service(HttpServletRequest req,HttpServletResponse res) throws

IOException {

int a = Integer.parseInt(req.getParameter("num1"));

int b = Integer.parseInt(req.getParameter("num2"));int c = a\*b;

res.setContentType("text/html");

PrintWriter out = res.getWriter();

out.println("Mulitplication of two numbers "+a+","+b+" = "+c);

}

}

**web.xml**

<servlet>

<servlet-name>first</servlet-name>

<servlet-class>Multiple</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>first</servlet-name>

<url-pattern>/Multiple.java</url-pattern>

</servlet-mapping>

**OUTPUT:**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**B)**

**AIM:** Create different web applications using servlets along with the database.

**PROGRAM:**

<html>

<body bgcolor="#FFFACD">

<h1 align="center">Registration Form</h1>

<form align="center" action="Resp.java" method="post">

Enter Name:<input type="text" name="uname"><br><br>

Enter Password:<input type="password" name="upass"><br><br>

<input type="submit">

<input type="reset">

</form>

</body>

</html>

**Resp.java**

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

import java.sql.\*;

public class Resp extends HttpServlet {

public void service (HttpServletRequest req,HttpServletResponse res) throws IOException

{

String param1 = req.getParameter("uname");

String param2 = req.getParameter("upass");

try{

PrintWriter pw = res.getWriter();

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/student","root","root");

String sql = "insert into form\_details values(?,?)";

PreparedStatement ps = con.prepareStatement(sql);

ps.setString(1,param1);

ps.setString(2,param2);

int rows = ps.executeUpdate();

if(rows>0) {

pw.println("Data Inserted Successfully");

}

con.close();

}

catch(Exception e) {

System.out.println(e);

}

}

}

**mysql**

create database student;

use student;

create table form\_details(name varchar(20),password varchar(30));

**web.xml**

<servlet>

<servlet-name>second</servlet-name>

<servlet-class>Resp</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>second</servlet-name>

<url-pattern>/Resp.java</url-pattern>

</servlet-mapping>

**OUTPUT:**

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, application

Description automatically generated

**EXPERIMENT-7**

**AIM:** Develop different web applications using JSP.

**DESCRIPTION:**

**JSP** technology is used to create web application just like Servlet technology. It can be

thought of as an extension to Servlet because it provides more functionality than servlet such

as expression language, JSTL, etc.

**JSP Scripting elements:**

The scripting elements provides the ability to insert java code inside the jsp. There are three

types of scripting elements:

* scriptlet tag
* expression tag
* declaration tag

JSP scriptlet tag

A scriptlet tag is used to execute java source code in JSP. Syntax is as follows:

<% java source code %>

**JSP expression tag :**

The code placed within **JSP expression tag** is written to the output stream of response.So you need not write out.print() to write data. It is mainly used to print the values of variable

or method.

Syntax of JSP expression tag

**<**%= statement %**>**

**JSP Declaration Tag:**

The **JSP declaration tag** is used to declare fields and methods.

The code written inside the jsp declaration tag is placed outside the service() method of auto

generated servlet.So it doesn't get memory at each request.

Syntax of JSP declaration tag

The syntax of the declaration tag is as follows:

**<**%! field or method declaration %**>**

**JSP request implicit object:**

The **JSP request** is an implicit object of type HttpServletRequest i.e. created for each jsp

request by the web container. It can be used to get request information such as parameter,

header information, remote address, server name, server port, content type, character

encoding etc.

String name=request.getParameter("uname");

**JSP response implicit object :**

In JSP, response is an implicit object of type HttpServletResponse. The instance of

HttpServletResponse is created by the web container for each jsp request.

response.sendRedirect("url");

**PROGRAM:**

**welcome.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"* pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>welcome page</title>

</head>

<body>

<%

String name=request.getParameter("uname");

String n\_ame=request.getParameter("u\_name");

**if**((name.equals("Snehita"))&&(n\_ame.equals("Snehita"))){

out.print("Welcome "+name);

}

**else**{

out.print("Enter a valid password");

}

%>

</body>

</html>

**details.html**

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>welcome page</title>

</head>

<body>

<form action=*"abc.jsp"*>

Enter your Name : <input type=*"text"* name=*"uname"*> </br>

Enter your Password : <input type=*"text"* name=*"u\_name"*>

<input type=*"submit"* value=*"go"*><br/>

</form>

</body>

</html>

**OUTPUT:**

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated