

28/02/2024

## Lab -07

- write an XML codes to validate various contents like XML, XSLT presentation and DTD usage

### → Student.xml

```
<!DOCTYPE student SYSTEM "student.dtd">
```

```
<Student>
```

```
  <Sname>Uday Kumar</Sname>
  <Spinno>0964</Spinno>
  <Sbranch>CSE</Sbranch>
```

```
</Student>
```

```
1  <!DOCTYPE student SYSTEM "student.dtd">
2  <student>
3    <sname>Uday Kumar</sname>
4    <spinno>0964</spinno>
5    <sbranch>CSE</sbranch>
6  </student>
7
```

Text Grid Schema WSDL XBRL Authentic Browser

student student\_info.xsl student\_xslt student\_xslt studentschema

Messages

1 File C:\Users\H-CSE-403-39\Downloads\XML\student\_xslt.xsl is valid.

VALID ✓

### → Student.dtd

```
<!ELEMENT student (sname,spinno,sbranch)>
<!ELEMENT sname (#PCDATA)>
<!ELEMENT spinno (#PCDATA)>
<!ELEMENT sbranch (#PCDATA)>
```

```
1 <!ELEMENT student (sname,spinno,sbranch)>
2 <!ELEMENT sname (#PCDATA)>
3 <!ELEMENT spinno (#PCDATA)>
4 <!ELEMENT sbranch (#PCDATA)>
```

Text Grid Schema WSDL XBRL Authentic Browser

student student\_info.xsl student\_xslt student\_xslt studentschema

Messages

1 File C:\Users\H-CSE-403-39\Downloads\XML\student\_xslt.xsl is valid.

VALID ✓

## → Student - info - xsit . xsl

```

<!DOCTYPE Student - info SYSTEM "student - xsit . dtd">
<?xml - Stylesheet type = "text / xsl" href = "student - xsit . xsl"?>
<student - info>
  <student>
    <sname> Uday kumar </sname>
    <spin> HU21CSEN0100964 </spin>
    <cgpa> 8.6 </cgpa>
  </student>
  <student>
    <sname> Karthik </sname>
    <spin> HU21CSEN0100735 </spin>
    <cgpa> 9.0 </cgpa>
  </student>
  <student>
    <sname> Kishore </sname>
    <spin> HU21CSEN010002 </spin>
    <cgpa> 7.9 </cgpa>
  </student>
</student - info>

```

```

1  <!DOCTYPE student - info SYSTEM "student - xsit . dtd">
2  <?xml - Stylesheet type = "text / xsl" href = "student - xsit . xsl"?>
3  <student - info>
4    <student>
5      <sname> Uday Kumar </sname>
6      <spin> HU21CSEN0100964 </spin>
7      <cgpa> 8.6 </cgpa>
8    </student>
9    <student>
10      <sname> Karthik </sname>
11      <spin> HU21CSEN0100735 </spin>
12      <cgpa> 9.0 </cgpa>
13    </student>
14    <student>
15      <sname> Kishore </sname>
16      <spin> HU21CSEN010002 </spin>
17      <cgpa> 7.9 </cgpa>
18    </student>
19  </student - info>

```

Text Grid Schema WSDL XSLT Authentic Browser  
 student student\_info.xsl student\_info.xsl student.xsl student.xsl student.xsl  
 Messages  
 File C:\Users\H-CSE-403-39\Downloads\xml\student\_xsit.xsl is valid.

Valid ✓

## → student - xsit . xsl

```

<xsl : stylesheet version = "1.0" xmlns : xsl = "http : //www.w3.org/1999/XSL/Transform">
  <xsl : template match = "/">
    <HTML>
      <BODY>
        <h1> student information </h1>
        <TABLE border = "1" bgColor = "pink">

```

```

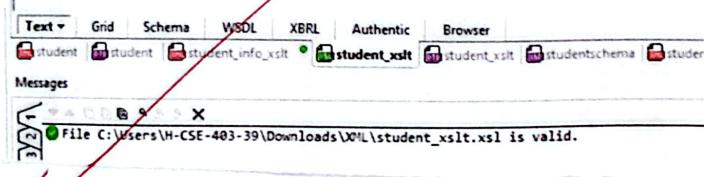
<tr>
    <xsl:value-of select = "sname" /></td>
    <xsl:value-of select = "spin" /></td>
    <xsl:value-of select = "cgpa" /></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

```

1 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
2   <xsl:template match="/">
3     <HTML>
4       <BODY>
5         <h1>student information</h1>
6         <TABLE border="1" bgcolor="pink">
7           <tr>
8             <th>STUDENTNAME</th>
9             <th>STUDENT PIN NUMBER</th>
10            <th>CGPA</th>
11          <xsl:for-each select="student-info/student">
12            <tr>
13              <xsl:value-of select="sname" /></td>
14              <xsl:value-of select="spin" /></td>
15              <xsl:value-of select="cgpa" /></td>
16            </tr>
17          </xsl:for-each>
18        </TABLE>
19      </BODY>
20    </HTML>
21  </xsl:template>
22</xsl:stylesheet>

```



valid ✓

→ So, in Browser side, the output is clearly displayed and it is valid, as shown in below output.

**student information**

STUDENTNAME	STUDENT PIN NUMBER	CGPA
Uday Kumar	HU21CSEN0100964	8.6
Karthik	HU21CSEN0100735	9.0
Kishore	HU21CSEN0100102	7.9

File C:\Users\H-CSE-403-39\Downloads\xml\student\_xslt.xsl is valid.

## → student-xslt.xls

```
<!ELEMENT Student-info (student+)>
```

```
<!ELEMENT Student (sname+, spin+, cgpa+)>
```

```
<!ELEMENT sname (#PCDATA)>
```

```
<!ELEMENT spin (#PCDATA)>
```

```
<!ELEMENT cgpa (#PCDATA)>
```

```
1 <!ELEMENT student-info (student+)>
2 <!ELEMENT student (sname+, spin+, cgpa+)>
3 <!ELEMENT sname (#PCDATA)>
4 <!ELEMENT spin (#PCDATA)>
5 <!ELEMENT cgpa (#PCDATA)>
```

Text Grid

student student\_info\_xslt student\_xslt student\_xslt studentschema students

Messages

File C:\Users\H-CSE-403-39\Downloads\XML\student\_xslt.xls is valid.

Valid ✓

## → StudentSchema

```
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema">
```

```
  <xss:element name = "student">
```

```
    <xss:complexType>
```

```
      <xss:sequence>
```

```
        <xss:element name = "sname" type = "xs:string"/>
```

```
        <xss:element name = "spinno" type = "xs:integer"/>
```

```
        <xss:element name = "sbranch" type = "xs:string"/>
```

```
      </xss:sequence>
```

```
    </xss:complexType>
```

```
  </xss:element>
```

```
</xss:schema>
```

```
1 <xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema">
2   <xss:element name="student">
3     <xss:complexType>
4       <xss:sequence>
5         <xss:element name="sname" type="xs:string"/>
6         <xss:element name="spinno" type="xs:integer"/>
7         <xss:element name="sbranch" type="xs:string"/>
8       </xss:sequence>
9     </xss:complexType>
10   </xss:element>
11 </xss:schema>
```

Text Grid Schema WSDL XBRL Authentic Browser

student student\_info\_xslt student\_xslt student\_xslt studentschema students

Messages

File C:\Users\H-CSE-403-39\Downloads\XML\student\_xslt.xls is valid.

Valid ✓

## → Students.xml

```
<student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="StudentSchema.xsd">  
    <sname> Kumar </sname>  
    <spinno> 789654 </spinno>  
    <sbranch> CSE </sbranch>  
</student>
```

```
1 <student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="studentSchema.xsd">  
2     <sname> Kumar </sname>  
3     <spinno> 789654 </spinno>  
4     <sbranch> CSE </sbranch>  
5 </student>  
6
```

File C:\Users\H-CSE-403-39\Downloads\VOL\student\_xslt.xls is valid.

valid ✓

B3  
7/3/29

08/03/2024

## Lab -08

- Deployment on Tomcat webserver

### Step 1: Setting up the path

- Go to settings
- click environment variables
- Set Path
- Edit Environment Variable
- c:\Program Files (x86)\Java\jdk-21\lib;
- c:\Program files(x86)\Java\jre-1.8\lib;
- c:\xampp\tomcat\lib\servlet-api.jar;

### Step 2: Open XAMPP Control panel

- click Start on Apache of port 80,443
- click Start on MySQL of port 3306
- Click start on Tomcat of port 8005,8080
- You can see Green Indication, which means the Control panel is set-up successfully.

### Step 3: Creating a Java file & class File Respectively

Coder → Import required java libraries

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

public class HelloWorld964 extends HttpServlet {
    private String message;
    public void init() throws ServletException {
        message = "Hello Uday Kumar how r u?";
    }
}
```

```

public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<h1>" + message + "</h1>");

}

public void destroy() {
    // do nothing
}
}

```

Now after saving this code, there will be a CLASS file created in the same folder which we have executed.

copy the file and paste it in a folder as mentioned below

- Open Local Disk (C)
- click on xampp folder
- press tomcat
- Go to webapps
- click on examples
- open WEB-INF
- In WEB-INF folder, you will find classes, click on it.
- paste the copied Class file in this folder.

To execute Java File , Here's the Terminal line code:

```

C:\Users\91778\Downloads\lab> javac -cp c:\xampp\tomcat\lib\servlet-api.jar -
c:\users\91778\downloads\lab\HelloWorld964.java

```

Which creates a CLASS FILE

#### Step 4 : Creating "web.xml" file

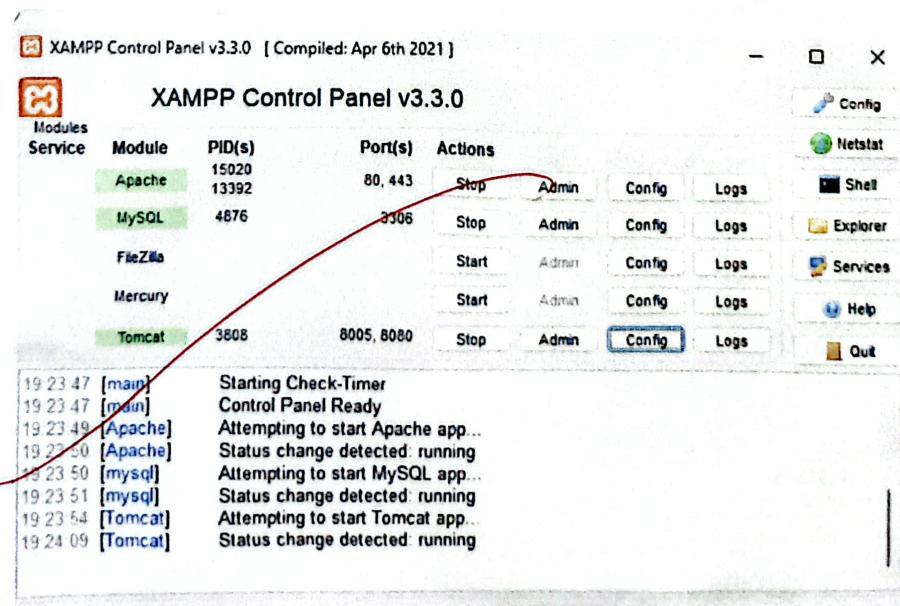
- open xampp control panel
- Beside tomcat , you can find stop,admin,config,logs
- click on config.
- you can find a file named "web.xml"
- click on that , which redirects to notepad file
- write the below code in that file

#### Code :

```
<servlet>
<Servlet-name>GITAM1 </Servlet-name>
<Servlet-class>HelloWorld964 </Servlet-class>
</servlet>

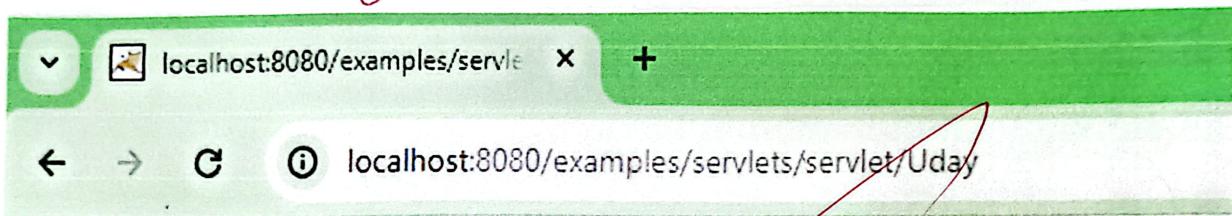
<servlet-mapping>
<Servlet-name>GITAM1 </Servlet-name>
<url-pattern>/servlets/servlet/uday </url-pattern>
</servlet-mapping>
```

This process is known as Servlet mapping.



## Step 5: Local Host verification

- Now Open the Web Browser
- click the link: localhost:8080/examples
- If this works, then you are in the right process
- click on "Servlet Examples"
- Now in Url, add localhost:8080/examples/servlets/servlet/Uday.
- This gives the final output as mentioned below.



**hello Uday Kumar how r u?**

BK  
1/3/24

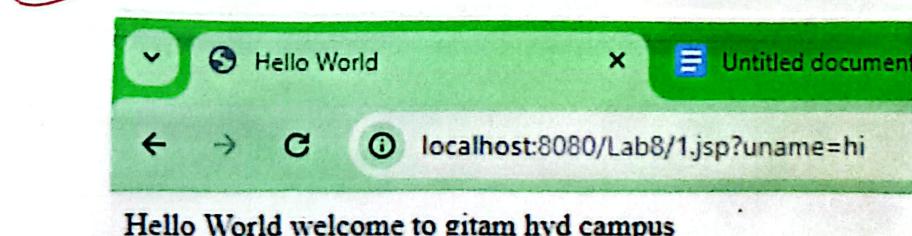
## ⇒ Java Servlet Pages (JSP) programs

- create a jsp form (jpfom.html)

```
<html>
<body>
<form action = "http://localhost:8080/Lab8/path.jsp">
<input type = "text" name = "uname">
<input type = "submit" value = "submit">
</form>
</body>
</html>
```

- create a jsp file (1.jsp)

```
<html>
<head>
    <title> Hello World </title>
<head>
<body>
    <%
        out.println("Hello world welcome to hyd campus");
    %>
</body>
</html>
```

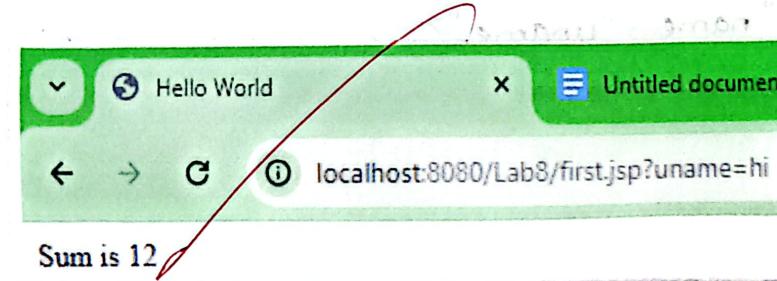


- Create a jsp file(first.jsp)

```

<html>
<head>
    <title> Hello World </title>
</head>
<body>
<%
    out.println("sum is");
    int x=5;
    int y=7;
    int z=x+y;
    out.println(z);
%>
</body>
</html>

```

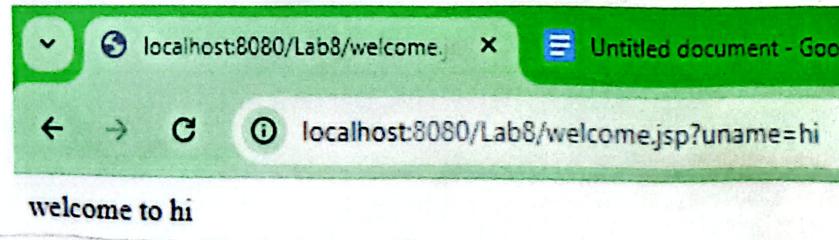


- Create a jsp file(welcome.jsp)

```

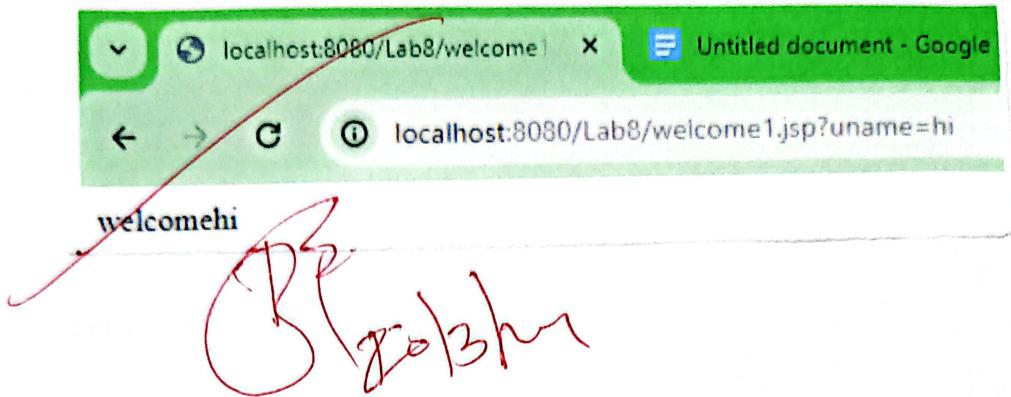
<html>
<body>
<%
    String name = request.getParameter("uname");
    out.println("Welcome to "+name);
%>
</body>
</html>

```



• create a jsp file (welcome1.jsp)

```
<html>  
<body>  
<% = "welcome" + request.getParameter("uname")%>  
</body>  
</html>
```



## Database Connectivity using Servlets

### Step 1 : Setup Environment Variable

- Download a file named : mysql-connector-java-5.1.12-bin.jar
- Go to Environment variables and create a class path and setup path

### Step 2 : Create "Register.java" and "Register.class"

- Create "Register.java" file
- Write the code mentioned below
- Register.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class extends HttpServlet {
    protected void doPost (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType ("text/html");
        PrintWriter out = response.getWriter();
        String fn = request.getParameter ("first");
        String ln = request.getParameter ("last");
        String em = request.getParameter ("email");
        String uid = request.getParameter ("uid");
        String pass = request.getParameter ("password");
        String cpass = request.getParameter ("confirm");
        String mob = request.getParameter ("mobile");
        try {
            Class.forName ("com.mysql.jdbc.Driver");

```

```
Connection con = DriverManager.getConnection
```

```
("jdbc:mysql://localhost:3306/gitam","root","");
```

```
PreparedStatement ps=con.prepareStatement
```

```
("insert into reg values (?, ?, ?, ?, ?, ?, ?, ?)");
```

```
ps.setString(1,fn);
```

```
ps.setString(2,ln);
```

```
ps.setString(3,em);
```

```
ps.setString(4,uid);
```

```
ps.setString(5,pass);
```

```
ps.setString(6,cpass);
```

```
ps.setString(7,mob);
```

```
int i=ps.executeUpdate();
```

```
if(i>0)
```

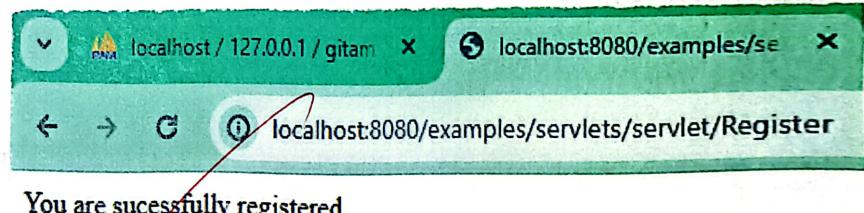
```
{ out.println("you are successfully Registered");}
```

```
}
```

```
}
```

```
Catch(exception se)
```

```
{ se.printStackTrace(); }
```



```
}
```

- Now execute the code and get "register.class" file.

- Deploy this file in xampp/examples/classes

Step 3 : Create "Retrieve.java" and "Retrieve.class"

- Create "Retrieve.java" file

- write the code mentioned below

- Retrieve.java :

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

```
import java.sql.*;
```

```
import javax.servlet.*;
```

```

public class Retrieve extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/gitam",
                "root", "sql");
            Statement stmt = c.createStatement();
            ResultSet rs = stmt.executeQuery("select first, last, email, uid, mobile from reg");
            out.println("<table border = \"1\"");
            out.println("<tr><td> First Name </td><td> Last Name </td><td> Email ID </td>
                <td> User ID </td><td> Mobile No </td></tr>");
            while (rs.next()) {
                out.println("<tr><td>" + rs.getString("first") + "</td><td>" + rs.getString("last") +
                    "</td><td>" + rs.getString("email") + "</td><td>" + rs.getString("uid") +
                    "</td><td>" + rs.getString("mobile") + "</td></tr>");
            }
            out.println("</table>");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}

```

Now execute the code and get "Retrieve.class" file

Deploy this file in xampp/examples/classes.

#### Step 4 : Create "Registration.html" file

- create a registration form file to fill the entries
- To create a registration file with validation, use lab05 validation code.

- Make sure to Add this statement in the Code

```
<body>
```

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

```
    <table align = "center">
```

```
        <form method = "post" action = "http://localhost:8080/examples/servlets/Servlet/Register">
```

#### Step 5 : Create "Login.html" file

- create a login form HTML file to retrieve action

- Login.html Code :

```
<html>
<head>
<meta charset = "ISO-8859-1">
<title> Insert Title Here </title>
</head>
<body>
    <h1> Login </h1>
```

```
    <form action = "http://localhost:8080/examples/servlets/servlet/Retrieve"
```

```
        "method = "post">
```

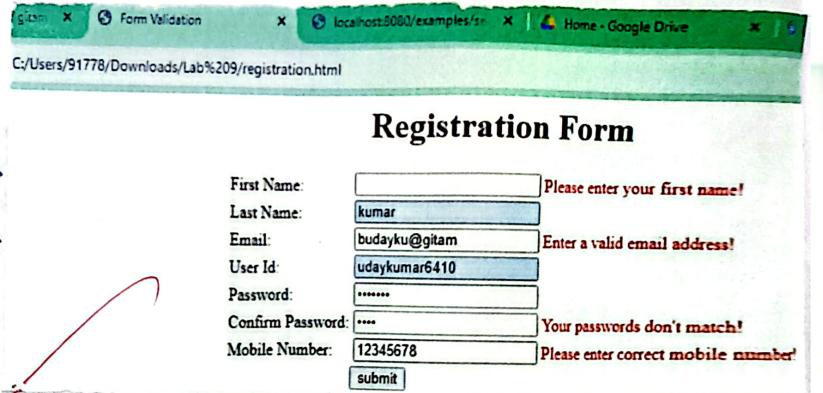
Enter username :  <br>

Enter password :  <br>

```
</form>
```

```
</body>
```

```
</html>
```



## Step 6 : Create a server in localhost:80 website

- Connect Apache, MySQL, Tomcat on XAMPP Control Panel

- Click on Browser

- Go to localhost:80 → Click phpMyAdmin

- Create a new database named "gitam"

- Create a table

```
create table reg(first varchar(20),  
last varchar(20), email varchar(30),  
uid varchar(20), pass varchar(20),  
cpass varchar(20), mobile varchar(20))
```

## Step 7 : Open Registration.html file in Browser and fill your details by

following validation rules

- After Submitting, you will get "you are successfully registered"

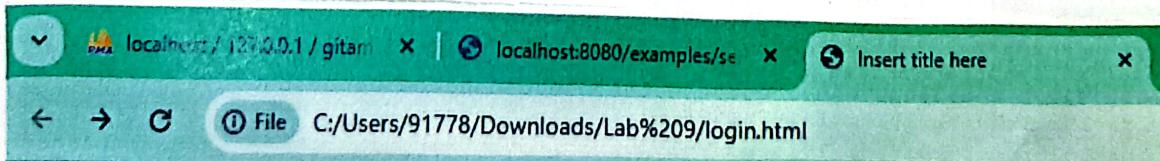
- Now check the details which we have filled on phpMyAdmin

- This is the registration

## Step 8 : To login and check details, click on "login.html" file

- Enter username and password

- Now the table with given entries are visible clearly.



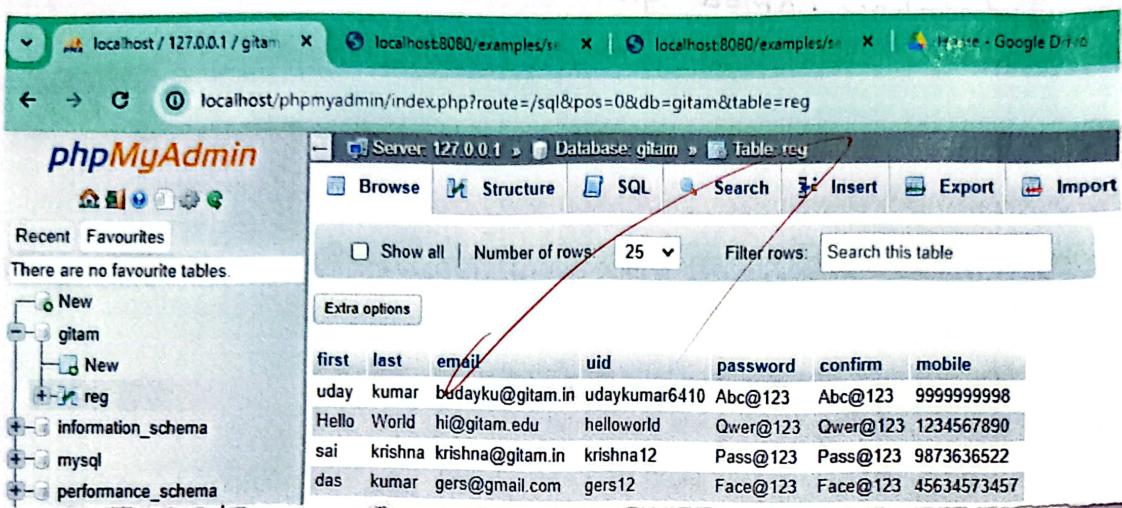
### Login

Enter username :

Enter password :

To view the registered details, click on localhost:80

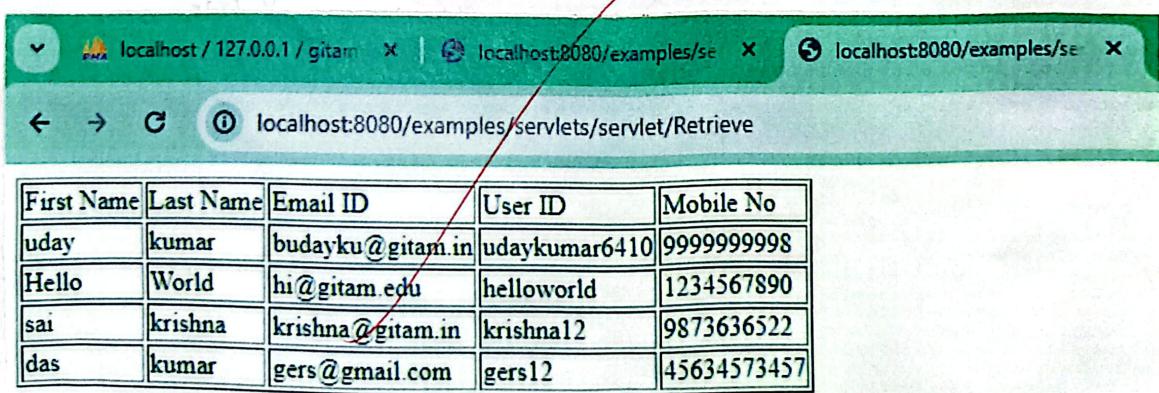
- Go to phpmyadmin
- open gitam database
- You can see the details as mentioned below



first	last	email	uid	password	confirm	mobile
uday	kumar	budayku@gitam.in	udaykumar6410	Abc@123	Abc@123	9999999998
Hello	World	hi@gitam.edu	helloworld	Qwer@123	Qwer@123	1234567890
sai	krishna	krishna@gitam.in	krishna12	Pass@123	Pass@123	9873636522
das	kumar	gers@gmail.com	gers12	Face@123	Face@123	45634573457

To retrieve the details in table form, run login.html file

- open login.html file
- Fill some details and submit
- You can view Table filled with registered details



First Name	Last Name	Email ID	User ID	Mobile No
uday	kumar	budayku@gitam.in	udaykumar6410	9999999998
Hello	World	hi@gitam.edu	helloworld	1234567890
sai	krishna	krishna@gitam.in	krishna12	9873636522
das	kumar	gers@gmail.com	gers12	45634573457

∴ Database using Servlets are executed successfully.

## Week 10

### Database Connectivity using jsp

#### Step 1 : Set up Environment Variable

- Download a file named : mysql-Connector-java-5.1.12-bin.jar
- Go to Environment variables and create a class path and setup path

#### Step 2 : Create a "registration.html"

- Go to xampp
- Open tomcat , click webapps
- Open examples and create a new folder "Database-using-jsp"
- Registration.html

```
<html><head><title>Registration Form</title></head><body><h1 align = "center">Registration Form</h1><form action = "reg.jsp" method = "post"><table align = "center"><tr><td>First Name :</td><td><input type = "text" name = "fname"/></td></tr><tr><td>Last name :</td><td><input type = "text" name = "lname"/></td></tr><tr><td>Email :</td><td><input type = "text" name = "email"/></td></tr><tr><td>User Name :</td><td><input type = "text" name = "userid"/></td></tr><tr><td>Password :</td><td><input type = "password" name = "pwd"/></td></tr><tr><td>Mobile No :</td><td><input type = "text" name = "mobile"/></td></tr><tr><td><input type = "submit" value = "submit"/></td><td><input type = "reset" value = "cancel"/></td></tr></table></form></body></html>
```

### Step 3 : Create a "reg.jsp" file

Enter the code given below

```

<%@ page import = "java.sql.*"%>
<%@ page import = "java.sql.*"%>
<%
String fname = request.getParameter ("fname");
String lname = request.getParameter ("lname");
String email = request.getParameter ("email");
String user = request.getParameter ("userid");
String pwd = request.getParameter ("pwd");
String mobile = request.getParameter ("mobile");

class.forName ("com.mysql.jdbc.Driver");
java.sql.Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306
gitam","root","");
Statement st = con.createStatement();

int i = st.executeUpdate ("insert into users values ('"+fname+"','"+lname+"','"+email+"'
"+user+"','"+pwd+"','"+mobile+"')");
if(i>0)
{
    out.println ("You are successfully Registered");
}
%>

```

### Step 4 : Create a "retrieve.jsp" file

Enter the code given below

```
<%@ page import = "java.sql.DriverManager"%>
```

```
<%@ page import = "java.sql.ResultSet"%>
```

```
<%@ page import = "java.sql.Statement"%>
```

```
<%@ page import = "java.sql.Connection"%>
```

**<h2 align = "center"><font><strong> Retrieve data from database in jsp </strong>
</font></h2>**

```
<table align="center" cellpadding="5" cellspacing="5" border="1">
<tr>
<tr>
<tr style="background-color: #A52A2A">
<th> First Name </th>
<th> Last Name </th>
<th> Email ID </th>
<th> User ID </th>
<th> Password </th>
<th> Mobile No </th>
</tr>
</tr>
try {
    String id = request.getParameter("userid");
    Class.forName("com.mysql.jdbc.Driver");
    java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/gitam", "root", "");
    Statement st = con.createStatement();
    String sql = "select * from users";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()) {
        <tr style="background-color: #DEBB87">
        <td><input type="checkbox" value="1" checked="checked" /></td>
        <td><input type="text" value="<%= rs.getString(1) %>" /></td>
        <td><input type="text" value="<%= rs.getString(2) %>" /></td>
        <td><input type="text" value="<%= rs.getString(3) %>" /></td>
        <td><input type="text" value="<%= rs.getString(4) %>" /></td>
        <td><input type="text" value="<%= rs.getString(5) %>" /></td>
        <td><input type="text" value="<%= rs.getString(6) %>" /></td>
        </tr>
    }
}
```

```
    }  
}  
  
catch (Exception e) {  
    e.printStackTrace();  
}  
}%>  
</table>
```

Step 5: make sure to place all these files in single folder.

I saved these files in

xampp > tomcat > webapps > examples > Database-using-jsp

Step 6: Now open xampp controller and Start connections APACHE, MySQL & tomcat

Step 7: open localhost:80

- click phpmyAdmin

- create a table named "users" in "gitam" database

→ To create table, use this code

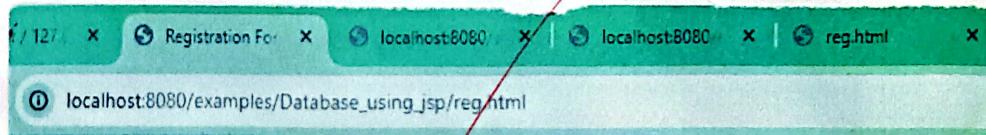
create table users {

    fname varchar (20), lname varchar (20),

    email varchar (30), userid varchar(20),

    password varchar (20), mobile varchar (20)

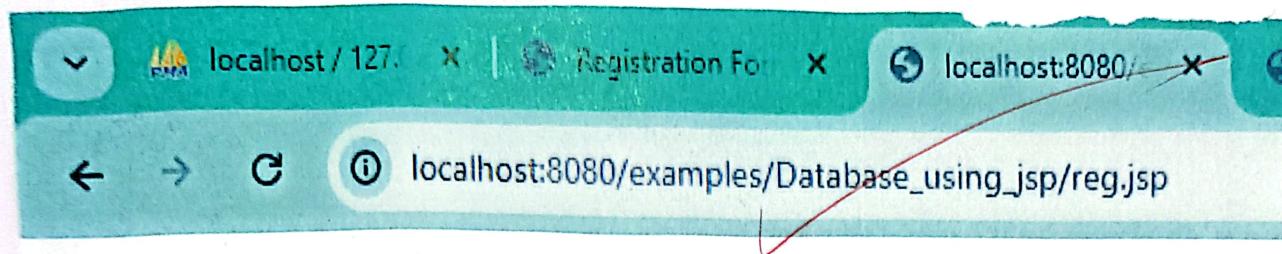
Step 8: open Registration.html file using URL mentioned below & fill details



## Registration Form

First name :	Bathula
Last name :	Uday Kumar
Email :	budayku@gitam.in
User name :	udaykumar6410
password :	.....
Mobile No:	1234567890
<input type="button" value="submit"/> <input type="button" value="cancel"/>	

Step 9 : After clicking on Submit, it will be redirected to "reg.jsp" and displays the message that you are successfully registered.



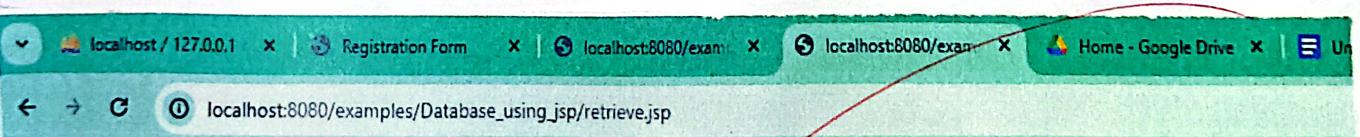
Step 10 : To check the registered details in gitam database , go to `localhost/phpMyAdmin /gitam/users`

A screenshot of the phpMyAdmin interface. On the left, the database schema is shown with the 'gitam' database selected. The 'users' table is highlighted. On the right, the 'users' table data is displayed in a grid:

fname	Iname	email	userid	password	mobile
Bathula	Uday Kumar	budayku@gitam.in	qqwqwr	1234	1234567890
Bathula	Uday Kumar	budayku@gitam.in	udaykumar6410	1234567	1234567890

Step 11 : To retrieve data from database in jsp , click URL

`localhost : 8080 /examples /Database -using -jsp /Retrieve.jsp`



Retrieve data from database in jsp

First Name	Last Name	Email ID	User ID	Password	Mobile No
Bathula	Uday Kumar	budayku@gitam.in	qqwqwr	1234	1234567890
Bathula	Uday Kumar	budayku@gitam.in	udaykumar6410	1234567	1234567890

Therefore, the data is successfully retrieved from the Database