

# **Department of Computer Science**

## *Gujarat University*



## *Certificate*

Roll No: 13

Seat No:   

*This is to certify that Mr./Ms. Antra koul student of MCA Semester – V has duly completed his/her term work for the semester ending in December 2020, in the subject of Web Technology towards partial fulfillment of his/her Degree of Masters in Computer Applications.*

10-12-2020  
*Date of Submission*

*Internal Faculty*

*Head of Department*

Department Of Computer Science  
Rollwala Computer Centre  
Gujarat University

MCA -

**Subject:** - Web Technology

**Name: - Antra koul**

**Roll No.: - 13**      **Exam Seat No.: -**

## Assignment 1:-

1. Write briefly about the different implicit objects in JSP.

*(Ans)* There are 9 implicit objects. These objects are created by the web container that are available to all the JSP pages.

1. Out:-

- For writing any data to the buffer.
- It is an object of PrintWriter.

In case of servlet we need to write:

PrintWriter out = response.getWriter();

- But in JSP, you don't need to write this code

Example:-

```
<% out.println("welcome!"); %>
```

2. Request:-

- It is an implicit object of HttpServletRequest.
- It can be used to get request information such as parameters, header information etc.
- It can be used to get, set and remove attributes from the JSP request scope.

Example:-

index.html :-

```
<form action = "welcome.jsp">  
<input type = "text" name = "uname">  
<input type = "submit" value = "go"> <br></form>
```

welcome.jsp

```
<%  
String name = request.getParameter("uname");
```

```
    out.println ("welcome " + name);  
%>
```

### 3) response:-

- In JSP, response is an implicit object of type `HttpServletResponse`. It is created by the web container for each JSP request.
- It can be used to add or manipulate response such as indirect response to another resource, send error etc.

example:-

```
welcome.jsp  
<%
```

```
    response.sendRedirect ("http://www.google.com");  
%>
```

### 4. Config:-

- It is an implicit object of type `ServletConfig`.
- It can be used to get initialization parameter for a particular JSP page.
- The config object is created by web container for each JSP page.
- It is used to get initialization parameters from the `web.xml`.

example:-

```
web.xml:-
```

```
<init-param>  
  <param-name> dname </param-name>  
  <param-value> Oracle </param-value>  
</init-param>
```

```
welcome.jsp:-
```

```
<% String driver = config.getInitParameter ("dname");  
%>
```

### 5) JSP application

- It is an implicit object of type ServletContext.
- The instance of ServletContext is created only once by the web container when application or project is deployed on the server.
- It can be used to get initialization parameter from configuration file (web.xml). It can also be used to get, set or remove attribute from the application scope.

Example :-

web.xml :-

```
<context-param>
  <param-name> dname </param-name>
  <param-value> sun.jdbc.odbc.JdbcOdbcDriver
</context-param>
```

welcome.jsp.

<%

```
%> String driver = application.getInitParameter("d-
  name");
```

### 6. Session.

- In JSP, session is an implicit object of type HttpSession.
- It can be used to set, get or remove attributes or to get session information.

example:-

index.html:-

```
<form action = "welcome.jsp">
  <input type = "text" name = "uname">
  <input type = "submit" value = "go"> <br/>
</form>
```

welcome.jsp:-

<%

```
String name = request.getParameter("uname");
out.print("Welcome " + name);
session.setAttribute("user", name);
<a href="second.jsp"> second.jsp </a>
```

%>

second.jsp

<%

```
String name = (String) session.getAttribute("user");
out.print("Hello " + name);
```

%>

#### 7. pageContext.

- pageContext is an implicit object of type PageContext.
- It can be used to get, set or remove attribute from one of the following scopes:-

1) page

2) request

3) session

4) application

example:-

welcome.jsp:-

<%

```
String name = request.getParameter("uname");
out.print("Welcome " + name);
```

%>

### 8. page.

- page is an implicit object of type Object class.
- This object is assigned to the reference of auto generated servlet class. It is written as:-  
Object page = this;
- For using this object it must be cast to Servlet type! For example:-  
<%@ (HttpServlet) page.log ("message"); %>  
Since, it is of type Object it is less used because you can use this object directly in JSP.  
For example:-  
<% this.log ("message"); %>

### 9. exception -

- exception is an implicit object of java.lang.Throwable class.
- It can be used to print the exception.
- But it can only be used in error pages.

Example:-

example.jsp

<%@ page isErrorPage = "true" %>

Sorry following exception occurred : <% = exception %>

### Q. What is a JSP and the tags to work with the bean?

Ans: JSP is a technology for developing web pages that support dynamic content which helps developers embed java code in html pages by making use of special JSP tags.

Using JSP, we can collect inputs from users

through web forms, present records from a database or another source, and create web pages dynamically.

Beans:-

- The `<jsp:useBean` action tag is used to locate or instantiate a bean class. If bean object of the bean class is already created, it doesn't create the bean depending on the scope. But if object of bean is not created, it instantiates the bean.

Syntax:-

```
<jsp:useBean id = "instanceName" scope = "page|request|session|application"
              class = "packageName.className"
              type = "packageName.className" beanName = "p-
              ackageName.className" <%= expression %> >
```

Attributes:-

- Id :- is used to identify the bean in the specified scope
  - Scope :- represents the scope of the bean. The default is page.
  - Class :- instantiates the specified bean class but it must have no arg. constructor.
  - Type :- provides the bean a data type.
  - BeanName :- instantiates the bean using `java.beans.Beans.instantiate` method () .
- `jsp:setProperty` and `jsp:getProperty` action tags:-

- The `setProperty` and `getProperty` action tags are used for developing web ~~pages~~ applications with Java bean.

- In web development, bean class is mostly used because it is a reusable software component that represents data.
- The `<jsp:setProperty` tag sets a property value or values in a bean using the setter method.
- Syntax:-

```
<jsp:setProperty name = "instanceOfBean" property = "xyz"
property = "propertyName" param = "parameterName"
property = "propertyName" value = "String"
<%= expression %> y />
```

- The `<jsp:getProperty` action tag returns the value of property.
- Syntax:-

```
<jsp:getProperty name = "instanceOfBean" property =
"propertyName" />
```

Example:-

`TestBean.java`

```
package action;
public class TestBean {
    private String message = "";
    public String getMessage() {
        return message;
    }
    public void setMessage(String message) {
        this.message = message;
    }
}
```

```

main.jsp
<html>
  <head>
    <title> JavaBeans </title>
  </head>
  <body>
    <center>
      <h2> Using JavaBeans </h2>
      <jsp:useBean id="test" class="Action.testBean">
        <jsp:setProperty name="test" property="message"
        value="Welcome" />
      <p> Got Message :: </p>
      <jsp:getProperty name="test" property="message">
    </center>
  </body>
</html>

```

Q. What are the different scopes of the beans created?  
Explain with example?

Scopes:-

1. page:-  
• 'page' scope means, the jsp object can be accessed from within in the page only where it was created.
- The default scope for jsp objects created using `<jsp:useBean>` tag is page.

2. request:-

- a jsp object created using the 'request' scope can be accessed from any page that serves the request.
- More than one page can serve that request.
- It will be bound to the request object.

- Session.
- Session Scope means the JSP object is accessible from pages that belong to the same session from where it was created.  
This JSP object that is created using the session scope is bound to the session object.
- application
- A JSP object created using application scope can be accessed from any page across the application.
- The JSP object is bound to application object

Example:-

Welcome form:-

```

<html>
  <head>
    <title> Welcome Form </title>
  </head>
  <body>
    <form action = "useBean.jsp" method = "post">
      ID: <input type = "text" name = "id" />
      Username: <input type = "text" name = "username" />
      Email: <input type = "email" name = "email" />
      Gender: <input type = "text" name = "gender" />
      <button> Session These </button>
    </form>
  </body>
</html>

```

Userbean.java

```
package user;
public class Userbean{
    private int id;
    private String username, email, gender;
    public int getId(){
        return id;
    }
```

```
    public void setId (int id){
    }
```

```
        this.id = id;
    }
```

```
    public String getUsername () {
        return username;
    }
```

```
    public void setUsername (String username) {
        this.username = username;
    }
```

```
    public void setEmail (String email) {
        this.email = email;
    }
```

```
    public String getEmail () {
        return email;
    }
```

```
    public void getGender () {
        return gender;
    }
```

```
    public void setGender (String gender) {
        this.gender = gender;
    }
```

```
}
```

## UseBean.jsp :-

```

<body>
  <jsp:usebean id="vObject" class="User.Usebean"
    scope="session"/>
  <a href="Sessionplace.jsp"> Click here </a>
  <jsp:setProperty property="u*" name="vObject">
</body>

```

## SessionPlace.jsp

```

<html>
  <head>
    <title> Information </title>
  </head>
  <body>
    <h1> Details are </h1>
    <jsp:usebean id="vObject" class="user.UseBean"
      &del scope="session" >
    <h3> UserID : <jsp:getProperty property="id"
      name="vObject"/>
    <h3> UserName : <jsp:getProperty property="username"
      name="vObject"/>
  </body>
</html>

```

4. Difference between JSP include directive and JSP include action tag.

### Include Directive

- It includes the file at translation time (the phase of JSP life cycle)

### Include Action

- It includes file at execution time

where JSP gets converted into equivalent servlet

2. If the included file is changed but not the JSP which is including it then the changes will reflect ~~not~~ <sup>only</sup> when we are using include directive as the JSP is not changed so it will not be translated.

The changes will be reflected only when we use JSP action tag.

### 3. Syntax:-

```
<%@ include file = "filename" %>
```

### Syntax:-

```
<@jsp:include page = "filename" >
```

### 4

In case of directive we cannot pass any parameter.

In action we can

pass parameters

```
<jsp:include page = "file" />
<jsp:param name = "paramname" value = "value" />
</jsp:include>
```

5.  
Ans:-

How are JSP's and servlets related

- JSP and servlets are both based on web Technologies.
- Both can be used to process dynamic content on server and return it to client using a typical web browser.
- JSP is internally compiled as to a servlet. As JSP tags can be understood directly. So the JSP tags are first parsed and a servlet file is created during runtime and request to that JSP is then handled by this generated servlet.

6. Give the types of JSP scripting elements.

JSP scripting elements:-

- The scripting element lets the ability to insert java code inside the JSP.
- There are 3 types of scripting elements
- 1. Scriptlet tag
- 2. Declarative tag
- 3. Expression tag.

1. Scriptlet tag:-

- It is used to execute Java code in JSP.

Syntax:-

<% java source code %>

Example:-

```

<html>
  <body>
    <% out.println ("Welcome to Jsp"); %>
  </body>
</html>

```

2.

### Expression Tag:-

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

#### Syntax:-

`<% = statement %>`

#### Example:-

```
<html>
  <body>
    Current time = <% = java.util.Calendar.get
    Instance().getTime() %>
  </body>
</html>
```

### 3. Declaration Tag:-

- It 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 does not get memory at each request.

#### Syntax:-

`<%! field or method declaration %>`

example:-

<html>

<body>

<?!

int cube(int n)

{

    return n\*n\*n;

}

?>

<? = "Cube of 3 is " + cube(3) ?>

</body>

</html>

## Assignment 2:-

1. Compare sendRedirect and forward method

### Forward

### SendRedirect

	Forward	SendRedirect
1.	When we use forward method request is transferred to another resource kept in the same server for further processing	In case of sendRedirect() request is transferred to another resource to different or domain or different server for further processing
2.	Web container handles all process internally and client or browser is not involved.	When you use sendRedirect container transfers the request to client or browser so url given inside the method is visible as a new request to client
3.	When forward is called on request dispatcher object we pass request and response object so one old request object is present on new resource which is going to process our request.	In case of sendRedirect call old request and response object is lost because it is treated as new request by browser.

4.	Visually we are not able to see forward address its transparent.	In address bar we are able to see the new redirected address its not transparent.
5.	It is faster.	It is slow because one extra round trip is required because completely new request is created and old request object is lost.
6.	When we redirect using forward and we want to use same data in new resource we can use request.setAttribute as we have request object available.	If we want to use or store the data in session or pass along the url.

F2. Discuss advantages of using JSP over Servlets

	JSP	Servlets
1.	Allows tag based programming. So extensive java knowledge is not required.	Does not allow tag based programming. So extensive java knowledge is required
2.	Suitable for both java se non java programmers	Not suitable for non java programmers

3. Use nine implicit objects which we can use directly in our program.
- Implicit objects are present but we can't use them directly. We need write additional code to use them.
4. Modification done in JSP program will be recognized by underlying server automatically without reloading of web server / application.
- Here we need to compile and reload manually.
5. Takes care of exception handling.
- Does not take care of exception handling. Programmers have to explicitly handle this.
6. Allow us to use separate presentation logic (HTML code) from Java code (business logic).
- Does not allow.
3. Discuss the attributes of page directives  
Page Directives:
- It provides attributes that gets applied to entire page. It defines page dependent attributes, such as scripting language, error page and buffering size.
  - It is used to provide instructions to container that pertains to current JSP page.

Syntax:-

<% @ page ..... %>

## Attributes:-

1. language:- It defines the programming language being used in the page.  
Ex:-  

$$<% @ page language = "java" %>$$
2. Extends:- This attribute is used to extend the classes like java does.  
Ex:-  

$$<% page extends = "demotest.DemoClass" %>$$
3. Import:- This attribute is mostly used attribute. It is used to tell the container to import other java classes, interfaces, enums etc while generating servlet code.  
Ex:-  

$$<% page import = "java.util.Date" %>$$
4. contentType:-  
  - It defines the character encoding scheme i.e it is used to set the content type and the character set of the response. Default is "text/html".  
Ex:-  

$$<% @ page contentType = "text/html" %>$$
5. info :- It defines a string which can be accessed by getServletInfo() method. It is used to get servlet description.  
Ex:-  

$$<%@ page info = "Servlet Info" %>$$

6. Session:- JSP creates a session by default. Sometimes we don't want a session to be created. When it is set to false, then we indicate the compiler to not create a session by default.

`<%@ page session = "true/false" %>`

7. isThreadSafe:- It defines threading model for the generated servlet. It indicates the level of thread safety implemented in the page. Its default value is true so simultaneous. We can use this attribute to implement single thread model.

`<%@ page isThreadSafe = "true/false" %>`

8. autoFlush:- This attribute specifies that the buffered outputs should be automatically flushed or not. Default value is true.

`<%@ page autoFlush = "true/false" %>`

9. Buffer:- The value represents the size of buffer. If there is no buffer then we can write as none. Default is 8KB.

`ex:-`

`<%@ page buffer = "16KB" %>`

10. isErrorPage:-

- It indicates that JSP page that has an error page will be checked in another JSP.
- Exceptions are available to this page only.

- default is false  
ex:-

`<%@ page isErrorPage = "true" %>`

11. Page Encoding :- It defines character encoding for JSP page. The default is "ISO-8859-1"  
ex:-

`<%@page pageEncoding = "ISO-8859-1" %>`

12. ErrorPage:- It is used to set error page for the JSP if throws an exception.  
ex:-

`<%@ page errorPage = "file1.jsp" %>`

13. isELIgnored:- It is flag attribute when we have to decide whether to ignore EL tags or not  
ex:-

`<%@ page isELIgnored = "true/false" %>`

14. Compare jsp:forward with jsp:include.

Property	include	forward
1. Deems	At translation time	At request time
2. Performance	comparatively slower	faster
3. No. of Servlets created	Only one	multiple

Property	include	forward
4. Response data sent.	Includes both jsp including and included.	Goes only that of included jsp.
5. Used when	when content of included file does not change often like advertisement banners.	when content of included file changes often.
6. Transfer of control	Execution control shifts temporarily to included file when it is exe. It transfers back.	Shifts permanently to the included file.
7. Response to clients.	Goes from same jsp which client requested.	Goes from diff event jsp.
8. Merge of response.	Both of including & included file.	No merging.
9. Extra activity.	When control returned including of other file can be done.	No extra activity.

5.

Explain `<jsp:fallback>` ?  
Syntax:-

`<jsp:fallback>` text message for user `</jsp:fallback>`

- A text message to display for the user if the plug-in cannot be started.
- If plug-in starts but the Applet or Bean does not, plug-in usually displays pop-up window explaining the error to user.
- Its used with `<jsp:plugin>` element

Example:-

```
<jsp:plugin type="applet" code="Code2.class".
codebase="applet" version="1.2" width="160"
height="150">
```

`<jsp:fallback>`  
plugin tag object or embed not supported  
by browser

`</jsp:fallback>`

`</jsp:plugin>`

6. What is bean? Explain in detail?

- A Java Bean is a class that should follow the following conventions:

1. It should have a no-arg constructor.
2. It should be serializable.
3. It should provide method to get and set the value of properties, known as getter setter methods.

How to access Java Bean class

```
package mypack;
```

```
public class Test {
```

```
    public static void main (String args [ ] ) {
```

```
        Employee e = new Employee (); // Object is  
        // created
```

```
        e.setName ("Antra"); // Setting value to the object
```

```
        System.out.print (e.getName ());
```

y

Advantages:-

- Java Bean property and methods can be exposed to another Application
- It provides an easiness to reuse software components

Disadvantages:-

- JavaBeans are mutable
- Creating getter & setter method for each property separately may lead to the boilerplate code.

7. What are JSP directives and how are they different from scripting elements?

- JSP directives are message to JSP container. They provide global information about an entire JSP page.
- JSP directives are used to give special instructions to a container for translation of JSP to servlet code.

- In JSP life cycle phase, JSP has to be converted to a servlet which is the translation phase.

- They give instructions to the container on how to handle certain aspects of jsp page processing.
- Directives can have many attributes by comma separated as key-value pairs.
- In jsp directive is described in <%@ %> tags.

Syntax:-

<%@ directive attribute = " " %>

Types:-

1. Page
2. Include
3. Taglib.

Compare with Scripting elements.

Scripting elements provides the ability to insert java code inside jsp whereas the jsp directives tell the container on how to actually handle different aspects of jsp page.

### Assignment 3:-

Q1:- What are the steps for implementing MVC?

Ans1- Steps:-

1. Define beans to represent data.
2. Use a servlet to handle request.
3. Populate the beans. The servlet invokes the business logic or data access code to obtain results. The results are placed in the beans that were designed in step 1.
4. Store the bean in the request, session or servlet context. The servlet calls .setAttribute on the request, session or servlet context object to store the reference to the beans that represent the result of a request.
5. Forward the request to a JSP page. The servlet determines which JSP page is appropriate to the situation and uses the forward method of RequestDispatcher to transfer control to that page.
6. Extract the data from the beans. The JSP page access bean with JSP:useBean and a scope matching the location of step 4. The page then uses JSP:getProperty to output the bean properties. The JSP page does not create or modify the bean, it merely extracts and displays the data that the servlet created.

Q2

Why MVC framework is required to build complex application?

Ans:-

Saves Time & effective use of Resources.

Due to the separation of components, MVC allows the reuse of business logic across platforms. In addition, multiple user interface can be developed in line without concerning the codebase. Two different programmers can work simultaneously on two different business logic. This makes the work faster, saves time, and helps in to manage the resources effectively.

Facilitates multiple views.

Due to the advantage of working on separate data and different business logic, duplication of code is certainly less. The separation of view model enables the user interface to display multiple views of the same data at the same time.

Modification does not effect entire model.

Change is part of life when it come to web application development changes become part of the ongoing development process. When it comes to user interface that changes are frequent may it be a change in colour. In addition UI updating can be made without changing dozen business logic.

## SEO friendly platform

Web development and SEO go hand in hand. MVC platform supports the development of SEO friendly applications. It provides ease to develop SEO friendly URLs in order to generate more visits on a specific page.

Q3 Discuss the operators available in EL.

Ans - JSP Expression Language (EL) supports most of the arithmetic and logical operators supported by Java.

S.NO	Operator & Description
1.	[ ] Access a bean or Map entry
2.	[ ] Access an array or list element
3.	( ) Group a subexpression to change the evaluation order
4.	+
	Addition
5.	-
	Subtraction
6.	*
	Multiplication

7. / or div  
Division
8. % or mod  
Modulo
9. == or eq  
Test for equality
10. != or ne  
Test for inequality
11. < or lt  
Test for less than
12. > or gt  
Test for greater than
13. <= or le  
Test for less than or equal
14. >= or ge  
Test for greater than or equal
15. && or and  
Test for logical AND
16. || or or  
Test for logical OR
17. ! or not  
Cinacy Boolean complement

18

empty

Test for empty variable values

- Q. How can you access a scoped variable using expression language? Example

Ans:-

1. PageScope :- Scoped variables from page.
2. requestScope :- Scoped variables from request scope
3. sessionScope :- Scoped variables from session scope
4. applicationScope :- Scoped variables from application scope.

The pageScope, requestScope, sessionScope and applicationScope variables provide access to variables stored at each scope level.

Application

SessionScope Example:-

In this example, attributes have been set using application implicit object and on the display page we have got those attributes using applicationScope of EL.

index.jsp

&lt;html&gt;

&lt;head&gt;

&lt;title&gt; EL example &lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;%

application.setAttribute("name", "Antra")

```

    application.setAttribute("rollno", "13");
%>
<a href = "display.jsp"> Click </a>
</body>
</html>

```

### display.jsp

```

<html>
<head>
<title> Display Page </title>
</head>
<body>
${applicationScope.name} <br>
${applicationScope.rollno} .
</body>
</html>

```

Q. How can you evaluate expression conditionally with EL?

#### 1. "JSP if-else"

"if else" is basic for all control flow statements, and it tells the program to execute the certain section of code only if the particular test evaluates to true.

- If the first condition is true then "if block" is executed and
- If it is false then "else block" is executed.

Syntax:-

```

if (test condition)
{
    // Block of statements
}
else
{
    // Block of statements
}

```

Example:-

```

<html>
    <head>
        <title> Conditional if-else </title>
    </head>
    <body>
        <%! int month = 5; %>
        <% if (month == 2) { %>
            <a>It's february </a>
        <% } else { %>
            <a>Any month </a>
        <% } %>
        </body>
    </html>.

```

## 2. JSP-Switch

The body of the switch ~~block~~ statement is called "switch block"

- It is used to check the no. of execution paths
- It can be used with all data types.
- It contains more than one cases and 1 default.
- It evaluates the exp. then executes all the statements following the matching case.

- Syntax:-

switch (operator)

{

case 1 :

Block of statement

break;

case 2 :-

Block of statement

⋮

default

Block of statements

break;

}

Example:-

<html>

<head>

<title> Switch </title>

</head>

<body>

<%! int week = 2; %>

<% switch (week) {

case 0 :

out.println ("Sunday");  
break;

case 1 :

out.println ("Monday");  
break;

case 2 :

out.println ("Tuesday");  
break;

default :

out.println ("Saturday");

7>  
 </body>  
 </html>

Q6:- Write the JSP comment syntax? -

A6:- JSP comments marks to test or statements that the JSP container should ignore.

Syntax:-

<%-- This is a JSP comment --%>

Example:-

```

<html>
  <head>
    <title> A Comment </title>
  </head>
  <body>
    <h2> A Test of Comments </h2>
    <%-- This is comment --%>
  </body>
</html>
  
```

Q7:- Explain the attributes of getProperty and setProperty tag of JSP?

SetProperty:-

- Sets a property value or values in a bean

Syntax:-

<jsp:setProperty

`name = "beanName".property = " " |`  
`property = "propertyName" [param = "parameterValue"]`  
`| property = "propertyName" value = "string" |`  
`<% = expression %> " />`

### Attributes:-

1. `name`:- The name of the instance of a bean that has already been created or located with `<jsp:useBean>` tag. The value of name must be same as the value of id in `<jsp:useBean>`. The `<jsp:useBean>` tag must be used before `<jsp:setProperty>`
2. `property = " "`:- Stores all the values in the request object parameters in the matching bean properties. The property names in the bean must match the request parameters.
3. `property = "propertyName" [param = "value"]`.  
Sets one Bean property to the value of request parameter. The request parameter can have a different name than Bean property.
4. `property = "propertyName" value = "string" | <% = expression %> "`.  
Sets one Bean property to a specific value. The value can be string or expression.

<ysp:getProperty>

Gets the value of a Bean property so that we can display in result page.

Syntax:-

<ysp:getProperty name = "beanInstanceName" property = "propertyName" />

Attributes:-

1. name = The name of instance.

2. property = The name of property whose value we want to get.

A

## Assignment-4

1. How to access nested bean and collection in JSP using JSTL?

**Dot Operator:-**

Also known as property access operator.  
This operator allows us to access property of any bean using dot notation.

For example:- To access the value of operator .property with name .property.name of bean , we can use below syntax .

`$[bean.property-name]`

There can be multiple level of nesting . EL allows to add scope to get the bean of a particular scope like `$[requestScope.bean.property]` . If we do not add scope, EL searches for the bean in the order of page, request, session, application

**Collection Access Operator:-**

EL supports `[]` operator which can be used to get the data from array and list along with beans .

For example:-

$\$$  { requestScope["a.b"] }  
 $\$$  { myList[1] } or  $\$$  { myList[4] }

Q. How to access body content of tag in JSTL?  
 For example:-

If our custom tag is xyz then we have to  
 access the content between

$<\!\!$ prefix:xyz>  
 Body of tag  
 $<\!\!$ prefix:xyz>

Details.java

```
package detail.com;
import java.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.io.*;
public class Details extends SimpleTagSupport {
  StringWriter sw = new StringWriter();
  public void doTag throws JspException, IOException {
    getJspBody().invoke(sw);
    JspWriter out = $getJspContext().getOut();
    out.println(sw.toString() + " Appended Custom
tag message");
  }
}
```

y

y

message.tld

```
<taglib>
  <tlib-version>1.0</tlib-version>
  <yjsp-version>2.0</yjsp-version>
  <short-name>My Custom Tag : MyMsg </short-name>
  <tag>
    <name>MyTagMsg</name>
    <tag-class>detail.Detail</tag-class>
    <body-content>scriptless</body-content>
  </tag>
</taglib>
```

index.jsp

```
<%@ taglib prefix = "myprefix" uri = "WEB-INF/
  message.tld" %>
<html>
  <head>
    <title>Body Content area</title>
  </head>
  <body>
    <myprefix:MyMsg>
      Test String
    </myprefix:MyMsg>
  </body>
</html>
```

## Assignment-5

Q1 Explain c:forEach, c:import and c:url with all the attributes.

Ans:- <c:forEach>

- This is used for executing the same set of statements for a finite number of times.
- It is similar to for loop in java.
- This is basically used for when we need to perform (execute) set of statements again and again for a specified number of times

Syntax:-

```
<c:forEach var = "counter" begin = "initial-value"
           end = "final-limit">
    // Block of statement
</c:forEach>
```

Attributes:-

Begin : The initial counter value

end : The final limit till which the loop executes

Var : counter variable name.

Example :-

```
<html>
  <head>
    <title> c:forEach </title>
  </head>
  <body>
    <c:forEach var = "counter" begin = "1" end = "10">
      <c:out value = "${counter}" />
    </c:forEach>
  </body>
</html>
```

<c:import>

- It is used for importing the content from another file/page to the current JSP page.

Syntax:-

<c:import var="variable-name" url="relative-url"/>

Attributes:-

url:- It is the url for address of file/page which needs to be imported

Var:- It is the variable which stores the data imported from another url.

Example:-

```
<html>
  <head>
    <title> c:import </title>
  </head>
  <body>
    <c:import var="mydata" url="/display.jsp">
    <clout value="<${mydata}"/>
    </c:import>
  </body>
</html>
```

<c:out>

- It is used for output formatting. It is mainly used when we want to open a jsp page based on the user input or based on the value of a variable.

Syntax:-

<c:out value="file1.jsp"/>

### Attributes:-

- var! - variable name to store the formatted url.
- context! - used for specifying the application.
- scope! - The scope in which the var attribute would be stored. It can be request, page, application or session.

### Example:-

```

<html>
  <head>
    <title> ${url} </title>
  </head>
  <body>
    <c:url value = "file1.jsp" />
  </body>
</html>

```

2. What is javascript? Explain its features.

### Javascript:-

- It is a programming language for web.
- It can update and change both HTML and CSS.
- It can calculate, manipulate and validate data.

### Features:-

- It is object based scripting language.
- Giving the user more control over the browser.
- It handles date & time.
- It can detect user's browser and OS.
- It is light weighted.
- It is a scripting language and it is not java.
- It is interpreter based scripting language.
- It is case sensitive.
- It provides pre-defined objects.

- Every statement in javascript must be terminated with semi colon

3. Describe advantages and disadvantages of javascript.

AS:- Advantages :-

1. Speed. Client side javascript is very fast because it can run immediately within the client-side browser. Unless outside resources are required, javascript is unhampered by network calls to backend server.
2. Simplicity. Javascript is relatively simple to learn and implement.
3. Popularity. Javascript is used everywhere on web.
4. Interoperability. Javascript plays nicely with other language and can be used in a huge variety of applications.
5. Server load. Being client-side reduces the demand on the website server.
6. Gives the ability to create rich interfaces.

Disadvantages :-

1. Client side security. Because the code extends on the user's computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable javascript.
2. Browser support. Javascript is sometimes interpreted differently by different browsers. This makes it somewhat difficult to write cross browser code.

4. How to create user defined object in javascript?  
Explain with example.

- Ans:-
- 1. Creating objects in javascript:-
  - 1. By object literal
  - 2. By creating instance of object directly using new keyword)
  - 3. By using an object constructor . Using new keyword.

1. By using object literal.

Syntax:-

```
object = { property1: value1, property2: value2...
           ... propertyN: valueN }
```

Example:-

```
<script>
    emp = { id: 102, name: "Shyam Kumar",
             salary: 40000 }
    document.write(emp.id + " " + emp.name +
                  " " + emp.salary);
</script>
```

2. By creating instance of Object

Syntax:-

```
var objectname = new Object();
```

Example:-

```
<script>
    var emp = new Object();
    emp.id = 101;
    emp.name = "Ravi Malik";
```

```

emp.salary = 50000;
document.write(emp.id + " " + emp.name + " " + emp.
salary);

```

&lt;/script&gt;

3.

By using constructor.  
Syntax:-

Object name = new object (values....)

- Here we need to create function with arguments. Each arguments value can be assigned in the current object by using this keyword.

- This keyword refers to current object.

Example:-

&lt;script&gt;

```
function emp(id, name, salary) {
```

```
this.id = id;
```

```
this.name = name;
```

```
this.salary = salary;
```

}

```
e = new emp(101, "Antra Meul", 30000);
```

```
document.write(e.id + " " + e.name + " " + e.salary);
```

&lt;/script&gt;

5. Difference between

Confirm Dialog box

Prompt Dialog box

i. It displays a pop up message with "Ok" & "cancel" buttons.

Prompt enables to take user's input with "Ok" and "cancel" buttons.

## confirm Dialog Box.

## Prompt Dialog Box

3. It is used to seek confirmation to proceed.

It returns value entered by user. It returns null if the user does not provide any input value.

## Include Directive

## Include Action

1. It includes file at translation time (the phase of JSP life cycle where the JSP gets converted into the equivalent servlet).

It includes file at runtime.

2. If the included file is changed but not the JSP which is including it then the changes will not be reflected.

The changes will only be reflected if we use include action.

## Syntax:-

<%@ include file = "file-name" %>

## Syntax:-

<%@ jsp:include page = "file-name" %>

4. It is not possible to pass parameters.

While using JSP-action include we can also pass parameters.

6. Define JSTL. State types of JSTL and discuss c:forTokens, c:redirect, c:catch, c:import and c:param.

Ans - JSTL:-

- It is a collection of useful JSP tags which encapsulates the core functionality common to many JSP applications.
- It has support for common, structural tasks etc.

Types:-

- Core Tags
- Formatting Tags
- SQL Tags
- XML Tags
- JSTL functions

1. <c:forTokens>

- Tag iterates over tokens which is separated by supplied delimiter. It is used to break string into tokens and iterate through each of tokens to generate output.

Example:-

```
<c:forTokens items = "Rahul-Nakul-Rayesh" delims =  
    "-"  
    var = "name">  
<c:out value = "${name}" /> <p>  
</c:forTokens>
```

2. <c:redirect>

- Tag redirects the browser to new url. It supports the context-relative URLs, and the <c:param>. It is used for redirecting the browser to an alternate URL by using automatic URL rewriting.

Example:-

```
<c:set var="url" value="0" scope="request"/>
<c:if test = "${!url}>
<c:redirect url="http://google.com"/>
</c:if>
```

### 3. <c:catch>

- It is used to catch any throwable exception that occurs in the body and optionally exposes it. In general it is used for error handling.

Example:-

```
<c:catch var="catch-theException">
<?out x=2/0;?>
</c:catch>
```

```
<c:if test = "${!catchtheException != null} y">
<p>The type of exception is : ${catchtheException}</p>
</c:if>
```

### 4. <c:import>

- It is used for including the content of any resources either within the server or outside the server.
- It provides all functionality of include action.

Example:-

```
<c:import var="data" url="http://www.google.com">
```

### 5. <c:param>

- It adds the parameter in a containing 'import' tag's URL. It allows the proper URL request parameter to be specified within URL and it automatically performs any necessary URL

encoding.

Example:-

```
<c:url value = "/index1.jsp" var = "complete URL"/>
<c:param name = "trackingId" value = "706"/>
<c:param name = "user" value = "Antra"/>
</c:url>
```

7. Explain JSP page directives import, session, buffer and errorPage attributes with example:-

1) Import:-

- This attribute is most used attribute in page directive.
- It is used to tell the container to import other java classes, interfaces, enums etc. while generating Servlet code. It is similar to import statement in java classes ; interfaces .

Syntax:-

```
c%@ page import = "value" %>
```

Examples:-

```
c%@ page import = "java.util.Date" %>
```

2) session.

- JSP creates session by default.
- Sometimes we don't want a session to be created in JSP and hence, we can set this attribute to false in that case. The default value of the session attribute is true.

Syntax:-

`<%@ page session = "true/false" %>`

Example:-

`<%@ page session = "false" %>`

### 3. Buffer

- Using this attribute the output response object may be buffered.
- We can define the size of buffering to be done using this attribute and default size is 8KB.
- It directs the servlet to write the buffer before writing to the response object.

Syntax:-

`<%@ page buffer = "value" %>`

Example:-

`<%@ page buffer = "16KB" %>`

### 4. errorPage

- This attribute is used to set the error page for the jsp page if jsp throws an exception and then it redirects to the exception page.

Syntax:-

`<%@ page ·errorPage = "value" %>`

Example:-

`<%@ page ·errorPage = "errorHandler.jsp" %>`

**DEPARTMENT OF COMPUTER SCIENCE  
ROLLWALA COMPUTER CENTRE  
GUJARAT UNIVERSITY  
M.C.A. -5**

ROLL NO : 13

**N A M E** : Antra koul

## **S U B J E C T :Web technology**

DEPARTMENT OF COMPUTER SCIENCE  
ROLLWALA COMPUTER CENTRE  
GUJARAT UNIVERSITY  
**M.C.A. -5**

ROLLNO : 13

NAME : Antra koul

SUBJECT : Web technology


### **Assignment1:**

#### 1. Basic Calculator

BasicCalculator.html

```
<html>
    <head>
        <title>
            Basic Calculator
        </title>
        <link rel="stylesheet" href="BasicCalculator.css">
        <script src="BasicCalculator.js" type="text/javascript"></script>
    </head>
    <body>
        <div align="center">
            <form>
                <input type="text" id="result"><br><br></form>
                <button class="button" value="C"
                    onclick="getExpression('C')">C</button>
                <button class="button" value="%"
                    onclick="getExpression('%')">%</button>
                <button class="button" value="+"
                    onclick="getExpression('+')">+</button>
                <button class="button" value="/"
                    onclick="getExpression('/')">/</button><br><br>
                <button class="button" value="7"
                    onclick="getExpression('7')">7</button>
                <button class="button" value="8"
                    onclick="getExpression('8')">8</button>
                <button class="button" value="9"
                    onclick="getExpression('9')">9</button>
                <button class="button" value="*"
                    onclick="getExpression('*')">*</button><br><br>
                <button class="button" value="4"
                    onclick="getExpression('4')">4</button>
                <button class="button" value="5"
                    onclick="getExpression('5')">5</button>
                <button class="button" value="6"
                    onclick="getExpression('6')">6</button>
                <button class="button" value="-" onclick="getExpression('-')">-
            </button><br><br>
```

```

                <button class="button" value="1"
onclick="getExpression('1')">1</button>
                <button class="button" value="2"
onclick="getExpression('2')">2</button>
                <button class="button" value="3"
onclick="getExpression('3')">3</button>
                <button class="button" value="."
onclick="getExpression('.')">.</button><br><br>
                <button class="button" value="0"
onclick="getExpression('0')">0</button>
                <button class="button" value="="
onclick="getExpression('=')">=</button>
            </form>
        </div>
    </body>
</html>
```

### BasicCalculator.js

```

function getExpression(e)
{
    var expression=sessionStorage.getItem("expression");
    var flag=sessionStorage.getItem("flag");
    var dot=sessionStorage.getItem("dot");
    if( expression == null){
        expression="";
        sessionStorage.setItem("expression",expression);
        sessionStorage.setItem("flag",0);
        sessionStorage.setItem("dot",0);
    }
    if(e== "+" || e== "-" || e== "%" || e== "/" || e== "*"){
        var count=sessionStorage.getItem("count");
        if(flag==0){
            alert("Wrong input!");
        }
        else{
            expression=expression.concat(e);
            sessionStorage.setItem("flag",0);
            sessionStorage.setItem("dot",0);
            sessionStorage.setItem("expression",expression);
            document.getElementById("result").value=expression;
        }
    }
}
```

```

        }
        else if(e=="."){
            if(dot==1){
                alert("Wrong input!");
            }
            else{
                expression=expression.concat(e);
                sessionStorage.setItem("expression",expression);
                sessionStorage.setItem("dot",1);
                document.getElementById("result").value=expression;
            }
        }
        else if( /\d/.test(e)){
            expression=expression.concat(e);
            sessionStorage.setItem("flag",1);
            sessionStorage.setItem("expression",expression);
            document.getElementById("result").value=expression;
        }
        else if(e=="=" && flag==0){
            alert("Wrong input!");
        }
        else if(e=="=" ){
            var result=eval(expression);
            document.getElementById("result").value=result;
            console.log(result);
            sessionStorage.setItem("expression","");
            sessionStorage.setItem("dot",0);
            sessionStorage.setItem("flag",0);
        }
        else if(e=="C"){
            sessionStorage.setItem("expression","");
            sessionStorage.setItem("flag",0);
            document.getElementById("result").value=null;
        }
    }
}

```

### BasicCalculator.css

```

.button {
    display: inline-block;
    padding: 15px 25px;
    font-size: 24px;
}

```

```
cursor: pointer;
text-align: center;
text-decoration: none;
outline: none;
color: #fff;
background-color: #4CAF50;
border: none;
border-radius: 15px;
box-shadow: 0 9px #999;
}

.button:hover {background-color: #3e8e41}

.button:active {
background-color: #3e8e41;
box-shadow: 0 5px #666;
transform: translateY(4px);
}

input[type=text] {
border: 2px solid #4CAF50;
border-radius: 4px;
width:250px;
height:50px;
}
```

“Output:”

Basic Calculator

File | D:/Semester-5/WebTechnology/Assignment1/Question%201/BasicCalculator.html

7+5

C % + /  
7 8 9 ×  
4 5 6 -  
1 2 3 .  
0 =

Type here to search

Basic Calculator

File | D:/Semester-5/WebTechnology/Assignment1/Question%201/BasicCalculator.html

16

C % + /  
7 8 9 ×  
4 5 6 -  
1 2 3 .  
0 =

## 2. Form Field validation

“FormValidation.html”

```
<html>
<head>
    <title>
        Form Validation
    </title>
    <link rel="stylesheet" href="Form.css">
    <script type="text/javascript" src="FormValidate.js"></script>
</head>
<body>
    <h3 align="center">Registration Form
    <table align="center">
        <form method="post" name="MyForm" onsubmit="return(validateForm());">
            <tr>
                <td>
                    <label for="name">Name:</label>
                </td>
                <td>
                    <input type="text" name="Name" >
                </td>
            </tr>
            <tr>
                <td>
                    <label for="contact_number">Contact number:</label>
                </td>
                <td>
                    <input type="text" name="Contact_number" maxlength="10">
                </td>
            </tr>
            <tr>
                <td>
                    <label for="email">E-mail:</label>
                </td>
                <td>
                    <input type="text" name="Email" id="Email">
                </td>
            </tr>
            <tr>
                <td>
                    <label for="password">Password:</label>
                </td>
                <td>
                    <input type="password" name="Password" id="password">
                </td>
            </tr>
        </form>
    </table>
</body>
```

```

        </td>
    </tr>
    <tr>
        <td>
            <input type="submit" value="submit">
        </td>
        <td>
            <input type="reset" value="reset">
        </td>
    </tr>
</form>
</table>
</body>
</html>

```

### "FormValidate.js"

```

function validateForm()
{
    if(document.MyForm.Name.value == "" || /\d/.test(document.MyForm.Name.value))
    {
        alert("Enter your name correctly!");
        document.MyForm.Name.focus();
        return false;
    }
    if(document.MyForm.Contact_number.value==""
    || /\D/.test(document.MyForm.Contact_number.value) ||
    (document.MyForm.Password.value).length<8){
        alert("Please provide your contact number!");
        document.MyForm.Contact_number.focus();
        return false;
    }
    if(document.MyForm.Email.value==""
    ||(document.MyForm.Email.value).indexOf("@")<1 ||
    ((document.MyForm.Email.value).indexOf("."))-
    ((document.MyForm.Email.value).indexOf("@"))<2){
        alert("Please provide your email!");
        document.MyForm.Email.focus();
        return false;
    }
    if(document.MyForm.Password.value==""
    ||(document.MyForm.Password.value).length<8 || /\s/.test(document.MyForm.Password.value)
    || /\W/.test(document.MyForm.Password.value)){

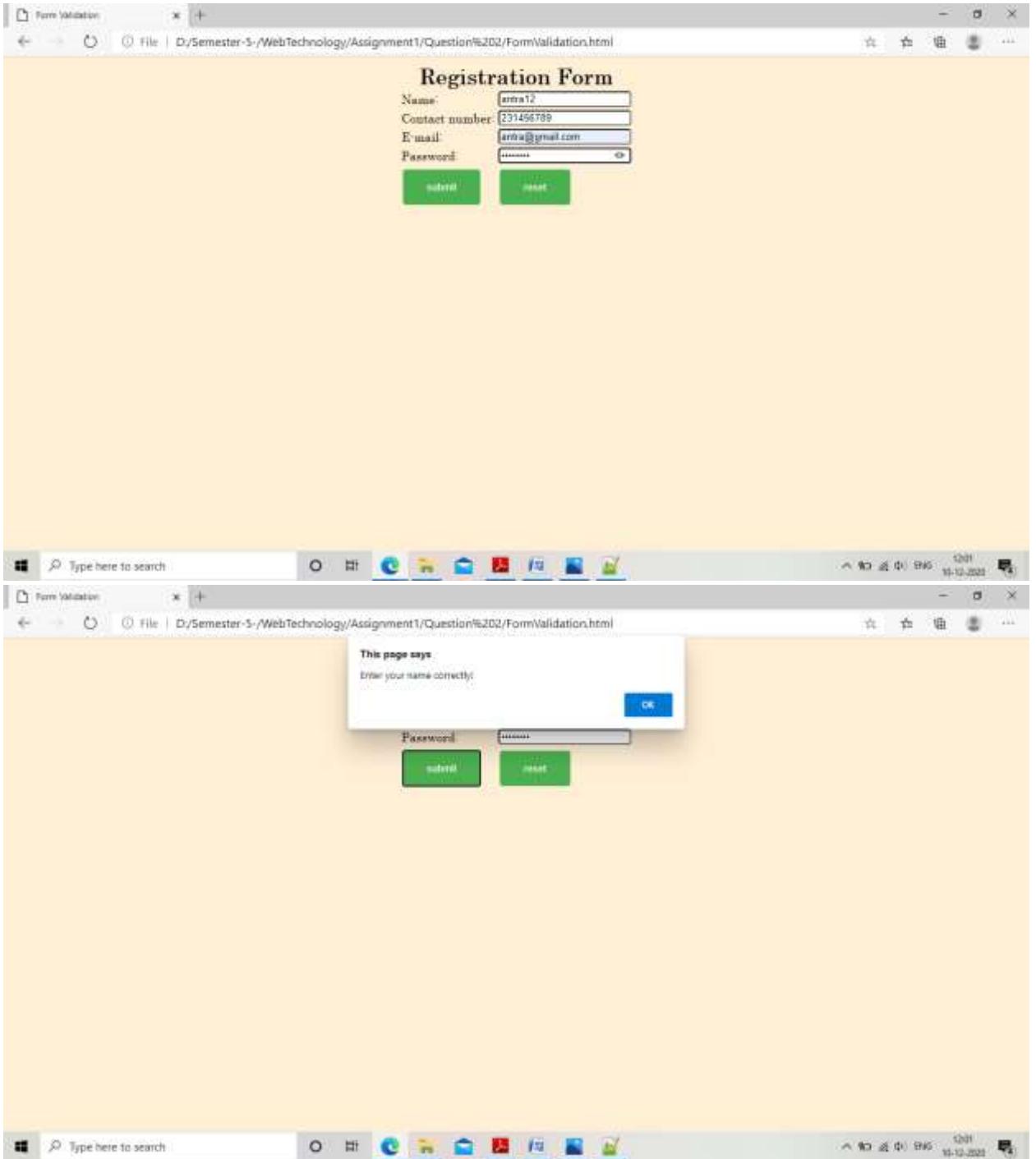
```

```
        alert("Please provide password!");
        document.MyForm.Password.focus();
        return false;
    }
    return true;
}
```

“Form.css”

```
body{
    background-color:papayawhip;
    font-family:Century;
    text-align:center;
    text-decoration:bold;
    text-indent:left;
    font-size:25px;
}
input{
    border-radius:4px;
    box-sizing:border-box;
}
input[type=button], input[type=submit], input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
}
```

“Output”



3. Image change on mouse over event

"ImageChange.html"

```
<html>
```

```
  <head>
```

```
<title>
    Image change
</title>

<script language="jacvascript" type="text/javascript">
var flag=0;

    function changelImage()
    {

        var imageSrc=["scenery1.jpg","scenery2.jpg"];

        if(flag==0)

        {

            document.getElementById("image").src=imageSrc[0];

            flag=1;

        }

        else

        {

            document.getElementById("image").src=imageSrc[1];

            flag=0;

        }

    }

</script>

</head>

<body>

<p align="center">

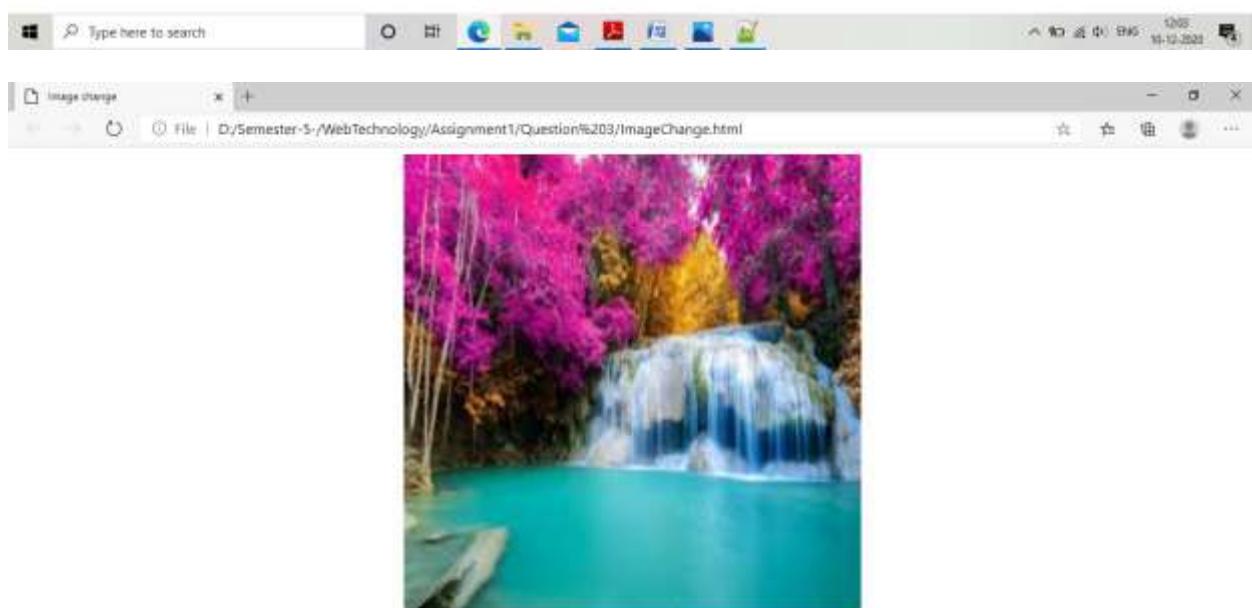


</p>
```

```
</body>
```

```
</html>
```

“Output”



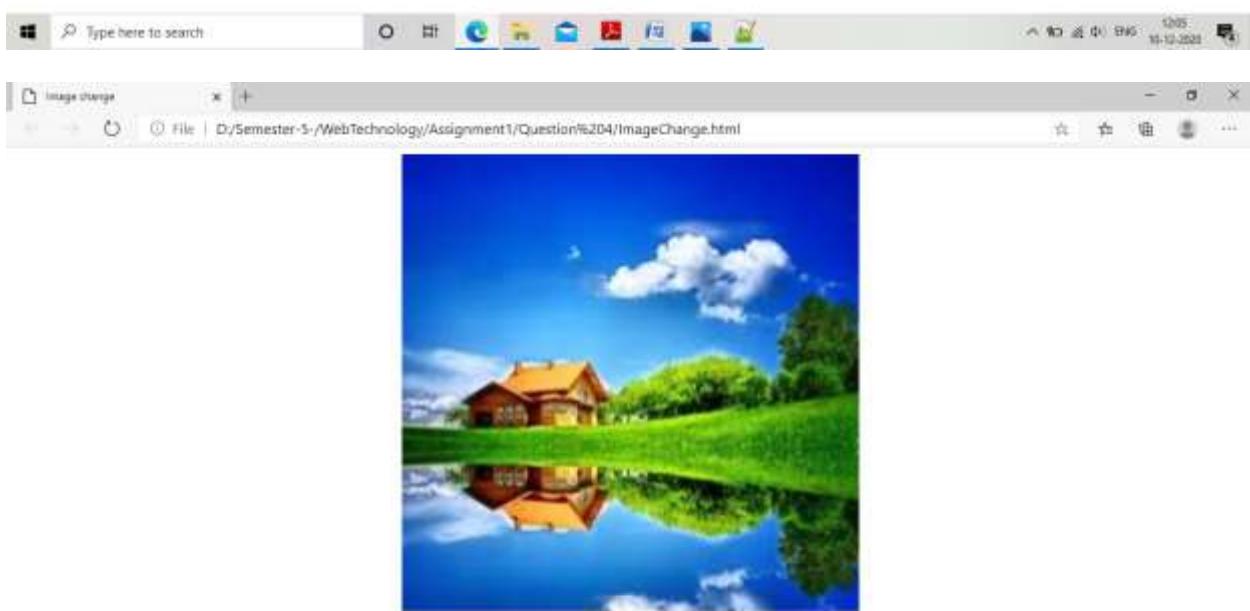
4.Random image change on mouse over event.

“ImageChange.html”

```
<html>

    <head>
        <title>
            Image change
        </title>
        <script language="jacvascript" type="text/javascript">
            function changeImage()
            {
                var number=Math.floor(Math.random() * 2);
                var imageSrc=["scenery1.jpg","scenery2.jpg","scenery3.jpg"];
                document.getElementById("image").src=imageSrc[number];
            }
        </script>
    </head>
    <body>
        <p align="center">
            
        </p>
    </body>
</html>
```

“Output”



5. Display your browser properties.

"BrowserProperties.html"

<html>

```
<head>
    <title>
        Browser Properties
    </title>
    <link rel="stylesheet" href="Table.css">
</head>
<body>
    <h2 align="center">Browser Properties</h2>
    <table align="center" >
        <tr>
            <td>
                Is cookie enabled?
            </td>
            <td id="cookie_enabled"></td>
            <script type="text/javascript" language="javascript">
                document.getElementById("cookie_enabled").innerHTML=
navigator.cookieEnabled;
            </script>
        </tr>
        <tr>
            <td>
                Application name
            </td>
            <td id="App_name"></td>
            <script type="text/javascript" language="javascript">
```

```
        document.getElementById("App_name").innerHTML=
navigator.appName;

</script>

</tr>

<tr>

<td>

Application code name

</td>

<td id="App_codename"></td>

<script type="text/javascript" language="javascript">

        document.getElementById("App_codename").innerHTML=
navigator.appCodeName;

</script>

</tr>

<tr>

<td>

Browser Engine name

</td>

<td id="browser_name"></td>

<script type="text/javascript" language="javascript">

        document.getElementById("browser_name").innerHTML=
navigator.product;

</script>

</tr>

<tr>

<td>

App version


```

```
</td>

<td id="App version"></td>

<script type="text/javascript" language="javascript">

    document.getElementById("App version").innerHTML=
navigator.appVersion;

</script>

</tr>

<tr>

<td>

        Browser Agent

</td>

<td id="browser_agent"></td>

<script type="text/javascript" language="javascript">

    document.getElementById("browser_agent").innerHTML=
navigator.userAgent;

</script>

</tr>

<tr>

<td>

        Browser Platform

</td>

<td id="browser_platform"></td>

<script type="text/javascript" language="javascript">

    document.getElementById("browser_platform").innerHTML=
navigator.platform;

</script>

</tr>
```

```
<tr>

    <td>

        Language

    </td>

    <td id="browser_language"></td>

    <script type="text/javascript" language="javascript">

        document.getElementById("browser_language").innerHTML= navigator.language;

    </script>

</tr>

<tr>

    <td>

        Is browser online?

    </td>

    <td id="browser_online"></td>

    <script type="text/javascript" language="javascript">

        document.getElementById("browser_online").innerHTML= navigator.onLine;

    </script>

</tr>

<tr>

    <td>

        Is browser java enabled?

    </td>

    <td id="java_enabled"></td>

    <script type="text/javascript" language="javascript">
```

```
document.getElementById("java_enabled").innerHTML=
navigator.javaEnabled();

</script>

</tr>

</table>

</body>

</html>
```

“Table.css”

```
th, td {

border-bottom: 1px solid #ddd;
padding: 15px;
text-align: left;
height: 50px;
vertical-align: bottom;

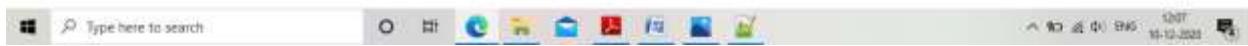
}

table {

border: 1px solid black;
}
```

“Output”

Browser Properties	
Is cookie enabled?	true
Application name	Netscape
Application code name	Mozilla
Browser Engine name	Gecko
App version	5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36 Edg/87.0.664.57
Browser Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36 Edg/87.0.664.57
Browser Platform	Win32
Language	en-US
Is browser online?	true
Is browser java enabled?	false



## 6. Open a new window on button click event

“NewWindow.html”

```
<html>
<head>
<title>
    New window
</title>
<link rel="stylesheet" href="Button.css">
<script type="text/javascript" language="javascript">
    function newWindow()
    {
        window.open();
    }
</script>
```

```
</head>

<body>

    <h3 align="center">Click the button to open a new window

    <table align="center">
        <form method="post">

            <tr>
                <td>
                    <button class="button" onclick="newWindow()">Click me!</button>
                </td>
            </tr>
        </form>
    </table>

</body>

</html>
```

“Button.css”

```
.button {

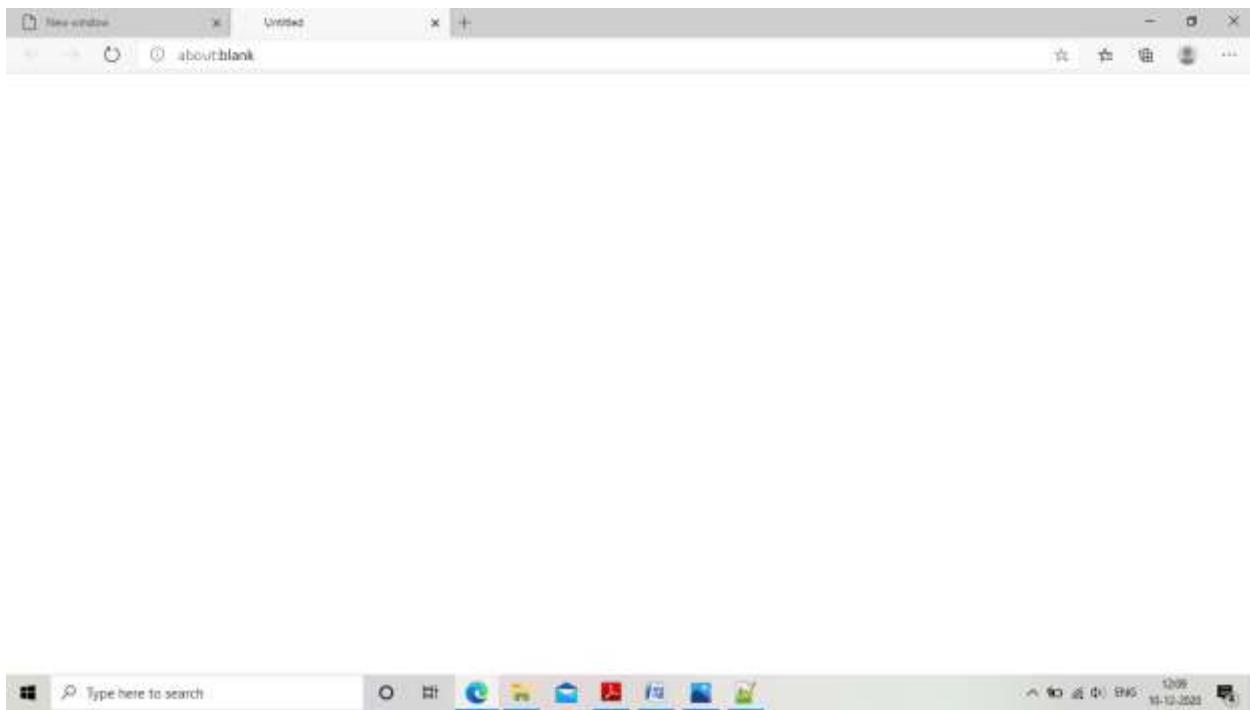
    display: inline-block;
    padding: 15px 25px;
    font-size: 24px;
    cursor: pointer;
    text-align: center;
    text-decoration: none;
    outline: none;
    color: #fff;
    background-color: #4CAF50;
```

```
border: none;  
border-radius: 15px;  
box-shadow: 0 9px #999;  
}  
  
.button:hover {background-color: #3e8e41}
```

```
.button:active {  
background-color: #3e8e41;  
box-shadow: 0 5px #666;  
transform: translateY(4px);  
}
```

“Output”





## 7. Move values from one list box to another

“listbox.html”

```
<html>

    <head>
        <title>
            Moving items to list box!
        </title>
        <link rel="stylesheet" href="ListBox.css">
    <script>
        function moveToSelected()
        {
            var x = document.getElementById("Fast_Food");
            var i=x.selectedIndex;
            var y=x.options[i].text;
            var option = document.createElement("option");

```

```
option.text =y;  
  
document.getElementById("selected_items").add(option);  
  
x.options.remove(i);  
  
}  
  
function moveToUnselected()  
  
{  
  
var x = document.getElementById("selected_items");  
  
var i=x.selectedIndex;  
  
var y=x.options[i].text;  
  
var option = document.createElement("option");  
  
option.text =y;  
  
document.getElementById("Fast_Food").add(option);  
  
x.options.remove(i);  
  
}  
  
  
 </script>  
 </head>  
 <body>  
 <h2 align="center" >  
 <form>  
 <select id="Fast_Food" class="myselect">  
 <option>Apple</option>  
 <option>Pear</option>  
 <option>Banana</option>  
 <option>Orange</option>
```

```

        </select>

    </form>

    <button class="button" id="button1" onclick="moveToSelected()">Move to
selected</button>

    <button class="button" id="button1" onclick="moveToUnselected()">Move to
unselected</button>

    <form>

        <select id="selected_items" class="myselect">

            <option></option>

        </select>

    </form>

</h2>

</body>

</html>

```

“ListBox.css”

```

body{

background-color:papayawhip;
font-family:Century;
text-align:center;
text-decoration:bold;
text-indent:left;
font-size:25px;

}

.myselect {
display: block;

```

```
font-size: 16px;  
font-family: sans-serif;  
font-weight: 700;  
color: #444;  
line-height: 1.3;  
padding: .6em 1.4em .5em .8em;  
width: 30%;  
max-width: 100%; /* useful when width is set to anything other than 100% */  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
-moz-appearance: none;  
-webkit-appearance: none;  
appearance: none;  
background-color: #fff;  
}  
  
}
```

```
.myselect::ms-expand {
```

```
display: none;
```

```
}
```

```
.myselect:hover {
```

```
border-color: #888;
```

```
}

/* Focus style */

.button {
    display: inline-block;
    padding: 15px 25px;
    font-size: 24px;
    cursor: pointer;
    text-align: center;
    text-decoration: none;
    outline: none;
    color: #fff;
    background-color: #4CAF50;
    border: none;
    border-radius: 15px;
    box-shadow: 0 9px #999;
}
```

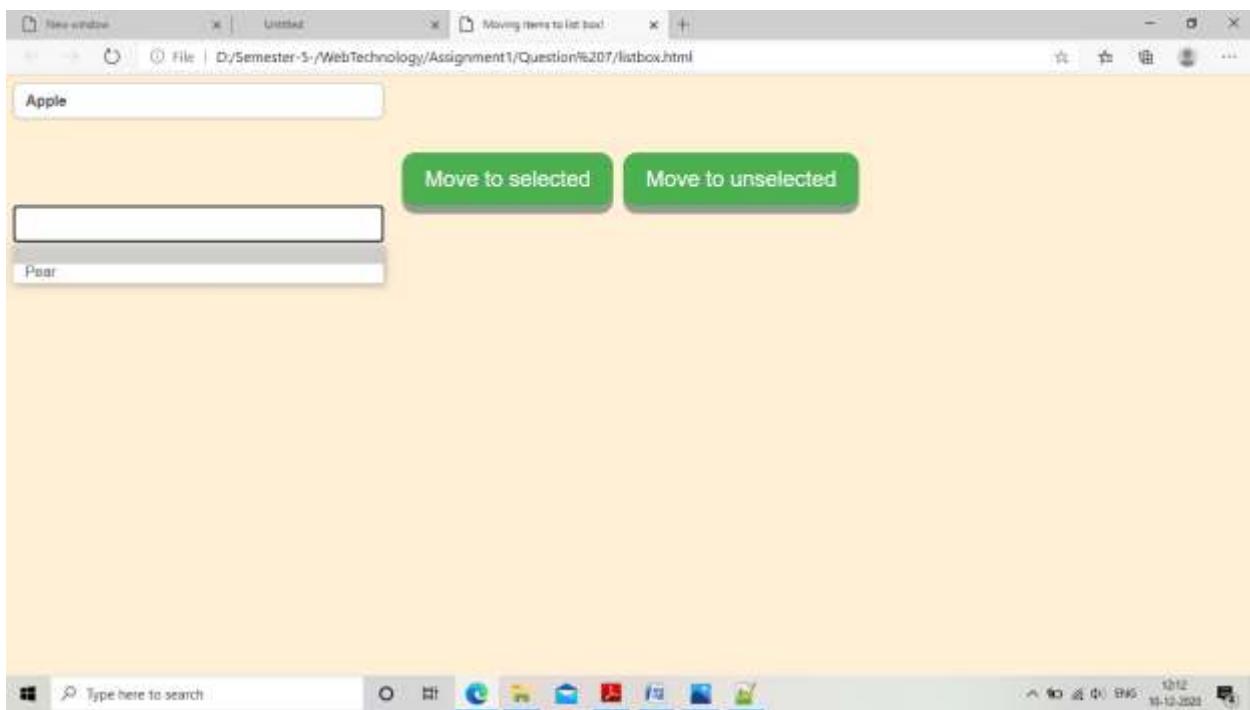
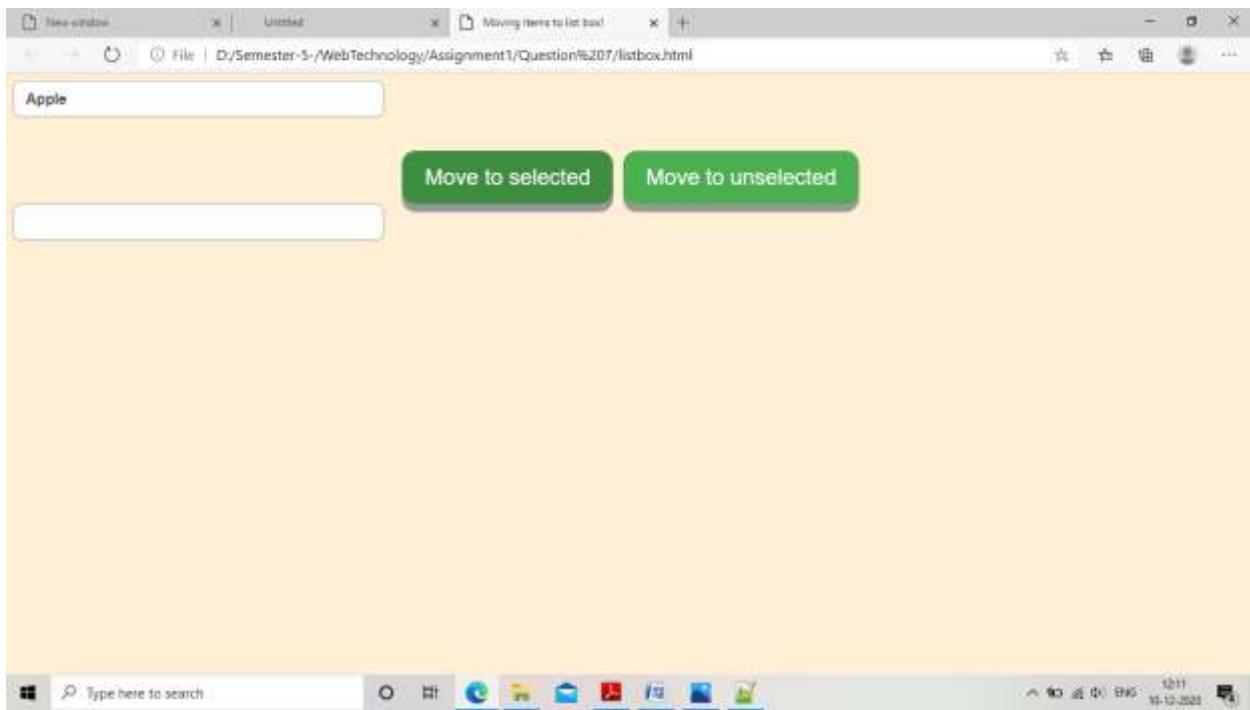
```
.button:hover {background-color: #3e8e41}
```

```
.button:active {
    background-color: #3e8e41;
    box-shadow: 0 5px #666;
    transform: translateY(4px);
}

.select {
```

```
display: block;  
font-size: 16px;  
font-family: sans-serif;  
font-weight: 700;  
color: #444;  
line-height: 1.3;  
padding: .6em 1.4em .5em .8em;  
width: 100%;  
max-width: 100%;  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
appearance: none;  
background-color: #fff;  
}
```

“Output”



8. Make country , state and city combo box and fill these boxes according to user choice

“ShowLocation.html”

<html>

```
<head>
    <title>
        Bill
    </title>
    <script src="Country.js" type="text/javascript"></script>
    <link rel="stylesheet" href="Country.css">
</head>
<body onload="printLocation()">
    Location
    <p id="location"></p>
</body>
</html>

"Country.html"

<html>
    <head>
        <title>
            Countries
        </title>
        <link rel="stylesheet" href="Country.css">
        <script type="text/javascript" src="Country.js"></script>
    </head>
    <body>
        <form>
            <label>Select country:</label>
            <select class="myselect" id="country" size="2" onchange="getStates()">
```

```

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

        <option value="Australia">Australia</option>

    </select>

</form>

<form>

    <label>Select state:</label>

    <select class="myselect" id="state" size="4" onchange="getCities()">

        </select>

</form>

<form action="ShowLocation.html">

    <label>Select city:</label>

    <select class="myselect" id="city" size="4" onchange="getLocation()">

        </select>

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

    <input type="reset" value="reset">

</form>

</body>

</html>

```

“Country.js”

```

function getStates()

{
    var India=["Gujarat","Delhi"];

    var Australia=["Tasmania","Victoria"];

    var states=new Map();

    states.set("India",India);

```

```
states.set("Australia",Australia);

var x=document.getElementById("country");

var i = x.selectedIndex;

var y = x.options[i].text;

var states_display = states.get(y);

for(var i=0;i<states_display.length;i++)

{

    var option=document.createElement("option");

    option.text=states_display[i];

    document.getElementById("state").add(option);

}

sessionStorage.setItem("location","Country:"+y+"<br>State:");

}

function getCities()

{

var Gujarat=["Ahmedabad","Surat"];

var Delhi=["Faridabad","Agra"];

var Tasmania=["Hobert","Burnie"];

var Victoria=["Ararat","Colac"];

var cities=new Map();

cities.set("Gujarat",Gujarat);

cities.set("Delhi",Delhi);

cities.set("Victoria",Victoria);

cities.set("Tasmania",Tasmania);

var x=document.getElementById("state");
```

```

var i = x.selectedIndex;

var y = x.options[i].text;

var cities_display = cities.get(y);

for(var i=0;i<cities_display.length;i++)

{

    var option=document.createElement("option");

    option.text=cities_display[i];

    document.getElementById("city").add(option);

}

var location=sessionStorage.getItem("location");

location=location+y;

sessionStorage.setItem("location",location+"<br>City");

}

function getLocation()

{

var x=document.getElementById("city");

var i = x.selectedIndex;

var y = x.options[i].text;

var location=sessionStorage.getItem("location");

location=location+y;

sessionStorage.setItem("location",location);

}

function printLocation(){

document.getElementById("location").innerHTML=sessionStorage.getItem("location");

}

```

“country.css”

```
body{  
    background-color:papayawhip;  
    font-family:Century;  
    text-decoration:bold;  
    text-indent:left;  
    font-size:20px;  
}  
  
.myselect {  
    display: block;  
    font-size: 16px;  
    font-family: sans-serif;  
    font-weight: 700;  
    color: #444;  
    line-height: 1.3;  
    padding: .6em 1.4em .5em .8em;  
    width: 30%;  
    max-width: 100%; /* useful when width is set to anything other than 100% */  
    box-sizing: border-box;  
    margin: 0;  
    border: 1px solid #aaa;  
    box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
    border-radius: .5em;  
    -moz-appearance: none;  
    -webkit-appearance: none;
```

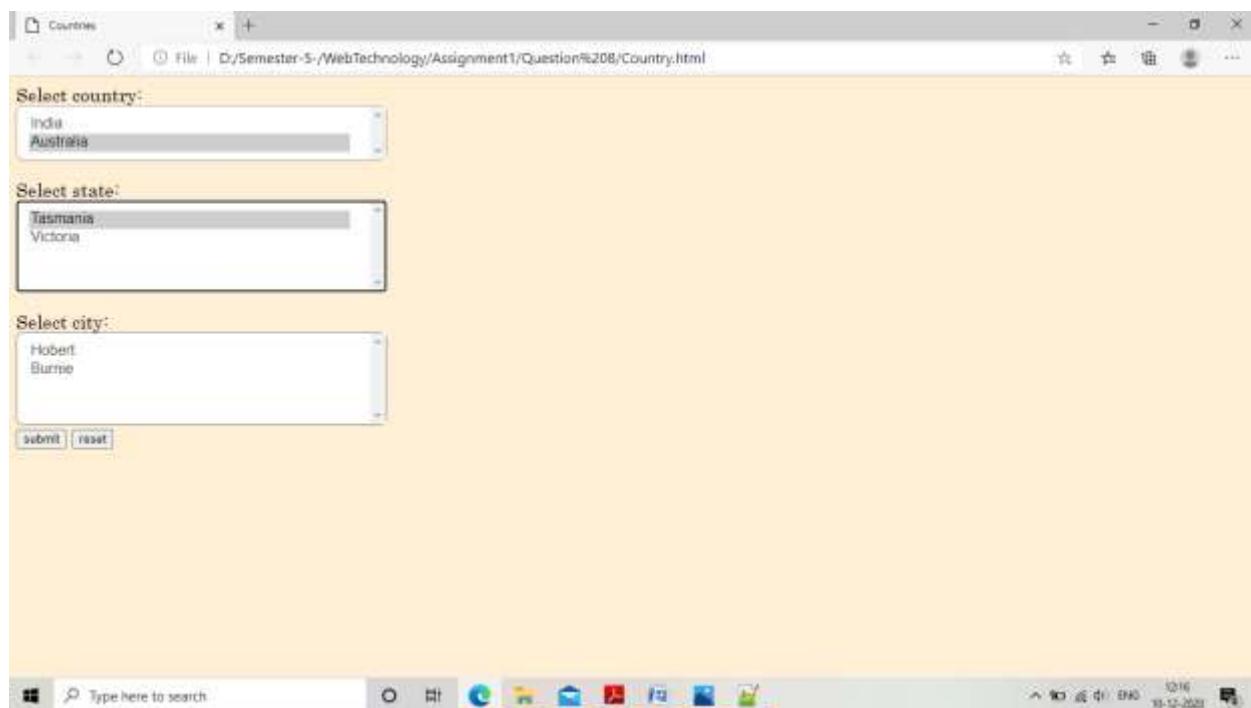
```
    appearance: none;  
    background-color: #fff;  
}  
  
.  
myselect::ms-expand {  
    display: none;  
}  
  
.  
myselect:hover {  
    border-color: #888;  
}  
  
/* Focus style */
```

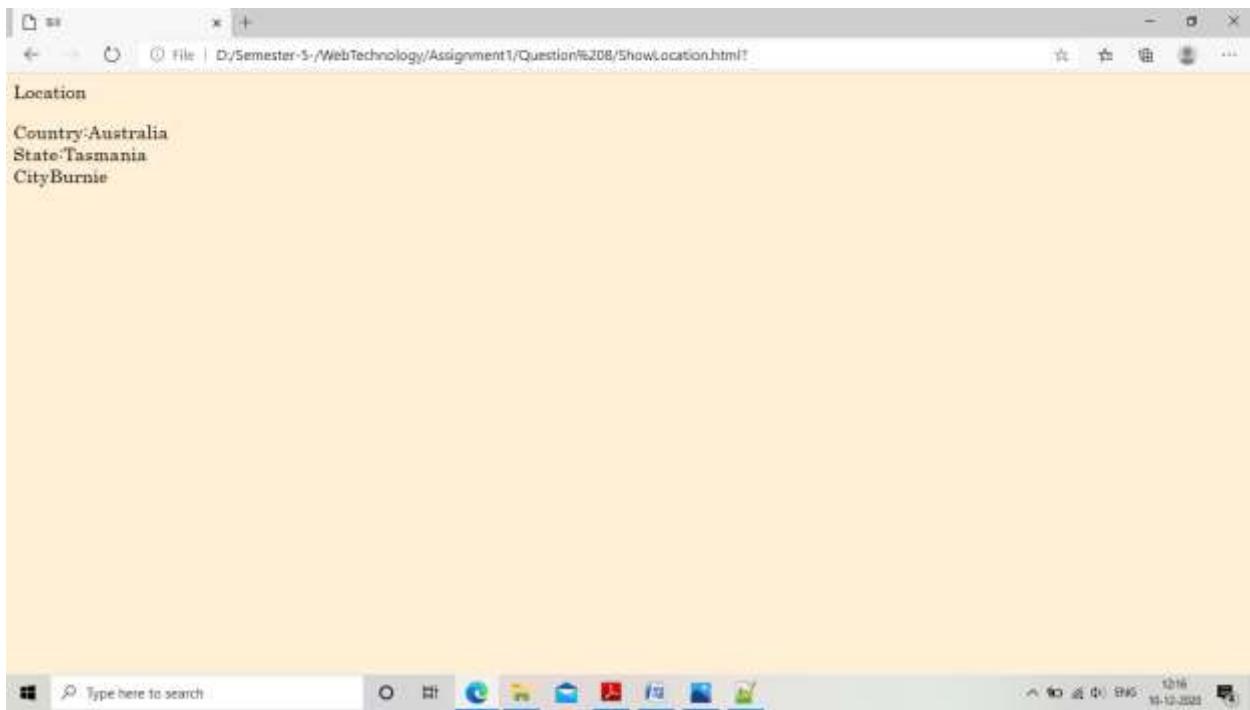
```
.button {  
    display: inline-block;  
    padding: 15px 25px;  
    font-size: 15px;  
    cursor: pointer;  
    text-align: center;  
    text-decoration: none;  
    outline: none;  
    color: #fff;  
    background-color: #4CAF50;  
    border: none;  
    border-radius: 10px;  
    box-shadow: 0 9px #999;
```

```
width:100px;  
height:50px;  
}  
  
.button:hover {background-color: #3e8e41}  
  
.button:active {  
background-color: #3e8e41;  
box-shadow: 0 5px #666;  
transform: translateY(4px);  
}  
  
input[type=text]{  
border-radius:4px;  
box-sizing:border-box;  
height:30px;  
}  
  
.select{  
display: block;  
font-size: 16px;  
font-family: sans-serif;  
font-weight: 700;  
color: #444;  
line-height: 1.3;  
padding: .6em 1.4em .5em .8em;  
width: 100%;
```

```
max-width: 100%;  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
appearance: none;  
background-color: #fff;  
}
```

“Output”





9. Display the list of food items and cold drinks and ask users to select the items and generate the bill.

“FoodItems.html”

```
<html>
  <head>
    <title>
      Food Items
    </title>
    <link rel="stylesheet" href="FoodItems.css">
    <script type="text/javascript" src="FoodItems.js"></script>
  </head>
  <body>

    <form >
      <label>Select colddrinks:</label>
```

```
<select class="myselect" id="colddrinks">

    <option value="Pepsi-Rs 50">Pepsi-Rs 50</option>

    <option value="Coke-Rs 50">Coke-Rs 50</option>

    <option value="ChaaS-Rs 30">ChaaS-Rs 30</option>

    <option value="Sprite-Rs 60">Sprite-Rs 60</option>

</select>

<label>Select food:</label>

<select class="myselect" id="fooditems">

    <option value="Pizza-Rs 100">Pizza-Rs 100</option>

    <option value="Pasta-Rs 50">Pasta-Rs 50</option>

    <option value="Brownie-Rs 80">Brownie-Rs 80</option>

    <option value="Pav baji-Rs 150">Pav baji-Rs 150</option>

</select>

</form>

<button class="button" onclick="moveSelectedColdDrink()">Add Coldrink</button>
<button class="button" onclick="moveSelectedFoodItems()">Add Fooditem</button>

<form action="Bill.html">

    <label>Selected coldrinks:</label>

    <select class="myselect" id="selected_colddrinks" size="4">

    </select>

    <label>Selected foodItems:</label>

    <select class="myselect" id="selected_fooditems" size="4">

    </select>

    <input type="submit" value="submit" class="button"
onclick="getBill()">
```

```
</form>

<button class="button" onclick="removeSelectedColdDrink()">Remove
Coldrink</button>

<button class="button" onclick="removeSelectedFoodItems()">Remove
Fooditem</button>

</body>

</html>
```

“FoodItems.js”

```
function moveSelectedColdDrink()

{

    var x=document.getElementById("colddrinks");

    var selected_index=x.selectedIndex;

    var option=document.createElement("option");

    option.text=x.options[selected_index].text;

    console.log(option.text);

    document.getElementById("selected_colddrinks").add(option);

}

function moveSelectedFoodItems()

{

    var x=document.getElementById("fooditems");

    var selected_index=x.selectedIndex;

    var option=document.createElement("option");

    option.text=x.options[selected_index].text;

    console.log(option.text);
```

```
document.getElementById("selected_fooditems").add(option);

}

function removeSelectedColdDrink()

{

var x = document.getElementById("selected_colddrinks");

var i=x.selectedIndex;

x.options.remove(i);

}

function removeSelectedFoodItems()

{

var x = document.getElementById("selected_fooditems");

var i=x.selectedIndex;

x.options.remove(i);

}

function getBill()

{

var total=0;

var bill="";

var colddrinks=document.getElementById("selected_colddrinks");

var fooditems=document.getElementById("selected_fooditems");

for(var i=0;i<colddrinks.length;i++){

    var item=colddrinks.options[i].text;

    bill=bill.concat(item);

    bill=bill+"<br>";

    total = total + parseInt(item.slice((item.indexOf("Rs"))+3,item.length));

}
```

```

}

bill=bill+"<br>";

for(var i=0;i<fooditems.length;i++){

    var item=fooditems.options[i].text;

    bill=bill.concat(item);

    bill=bill+"<br>";

    total = total + parseInt(item.slice((item.indexOf("Rs"))+3,item.length));

}

bill=bill+"<br>";

bill=bill+"Total:";

bill=bill+total;

sessionStorage.setItem("bill",bill);

}

function printBill()

{

    document.getElementById("Bill").innerHTML=sessionStorage.getItem("bill");

}

"Bill.html"

<html>

<head>

    <title>

        Bill

    </title>

    <script src="FoodItems.js" type="text/javascript"></script>

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

```

```
</head>

<body onload="printBill()">

    Bill:

    <p id="Bill"></p>

</body>

</html>
```

“FoodItems.css”

```
body{

    background-color:papayawhip;

    font-family:Century;

    text-decoration:bold;

    text-indent:left;

    font-size:20px;

}

.myselect {

    display: block;

    font-size: 16px;

    font-family: sans-serif;

    font-weight: 700;

    color: #444;

    line-height: 1.3;

    padding: .6em 1.4em .5em .8em;

    width: 30%;

    max-width: 100%; /* useful when width is set to anything other than 100% */

    box-sizing: border-box;
```

```
margin: 0;  
  
border: 1px solid #aaa;  
  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
  
border-radius: .5em;  
  
-moz-appearance: none;  
  
-webkit-appearance: none;  
  
appearance: none;  
  
background-color: #fff;  
  
}
```

```
.myselect::ms-expand {  
  
display: none;  
  
}
```

```
.myselect:hover {  
  
border-color: #888;  
  
}  
  
/* Focus style */  
  
.button {  
  
display: inline-block;  
  
padding: 15px 25px;  
  
font-size: 15px;  
  
cursor: pointer;  
  
text-align: center;  
  
text-decoration: none;
```

```
outline: none;  
color: #fff;  
background-color: #4CAF50;  
border: none;  
border-radius: 10px;  
box-shadow: 0 9px #999;  
width:100px;  
height:50px;  
}
```

```
.button:hover {background-color: #3e8e41}
```

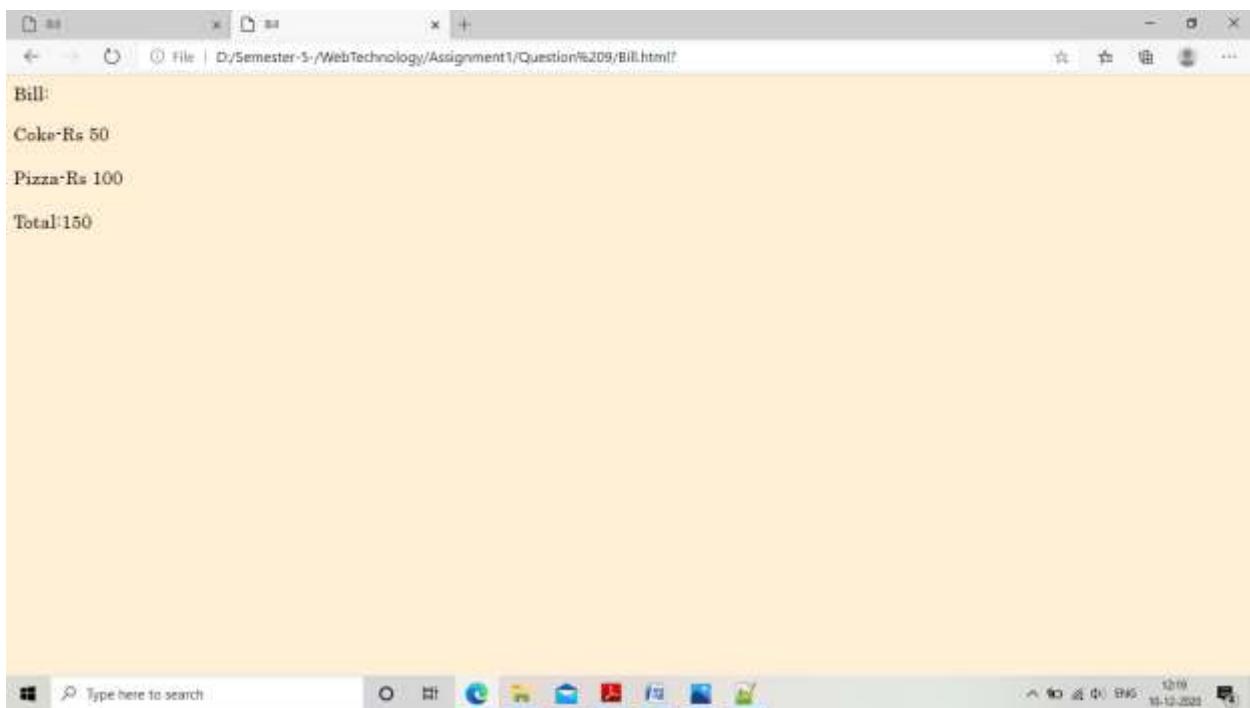
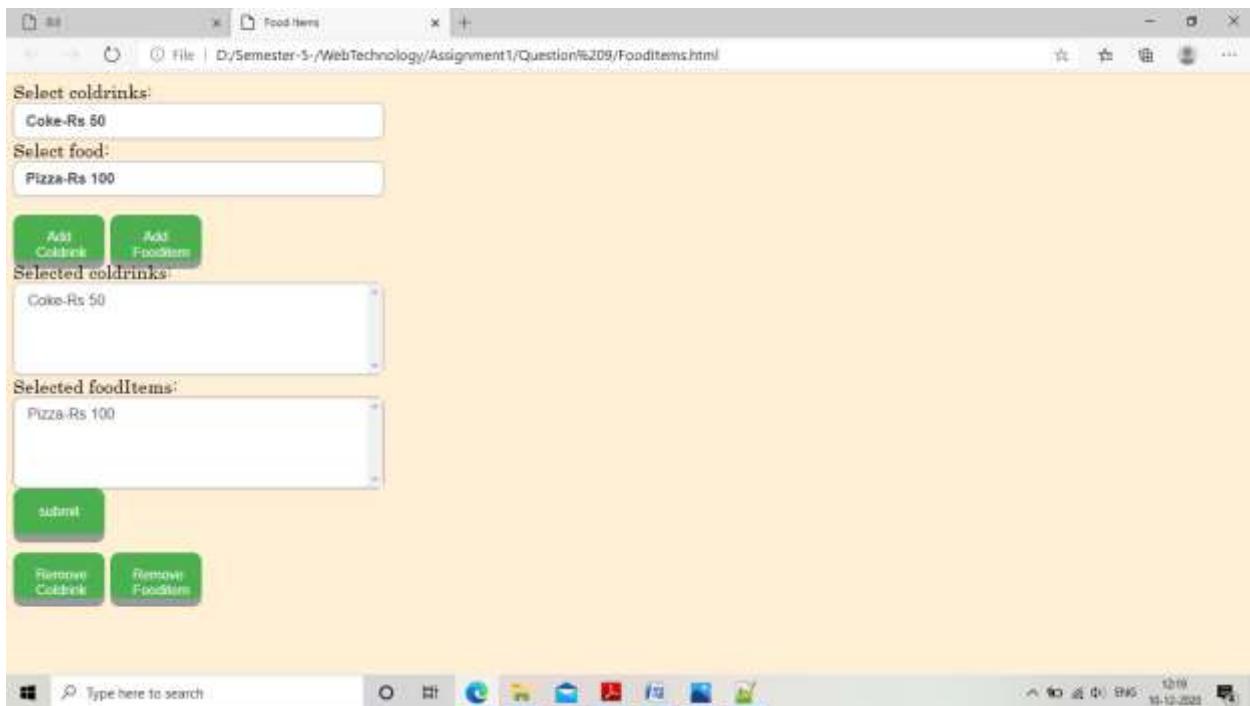
```
.button:active {  
background-color: #3e8e41;  
box-shadow: 0 5px #666;  
transform: translateY(4px);  
}
```

```
input[type=text]{  
border-radius:4px;  
box-sizing:border-box;  
height:30px;  
}
```

```
.select {  
display: block;  
font-size: 16px;
```

```
font-family: sans-serif;  
font-weight: 700;  
color: #444;  
line-height: 1.3;  
padding: .6em 1.4em .5em .8em;  
width: 100%;  
max-width: 100%;  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
appearance: none;  
background-color: #fff;  
}
```

“Output”



10. Display the current time and update it after every 1 sec.

“CurrentTime.html”

```
<html>
```

```
<head>
```

```
<title>
    Displaying current time
</title>

</head>

<body>
    1<h2><p align="center" id="time" style="font-family:Segoe Script;"></h2>
    </p>
    <script>
        window.onload=function() {
            getTime();
        }
        function getTime()
        {
            var today=new Date();
            var hrs=today.getHours();
            var min=today.getMinutes();
            var sec=today.getSeconds();

            min=formatDigits(min);
            sec=formatDigits(sec);

            document.getElementById("time").innerHTML=hrs+":"+min+":"+sec;
            setInterval("getTime()",1000);
        }
    
```

```
function formatDigits(digit)
{
    if(digit<10){

        digit="0"+digit;

    }

    return digit;
}

</script>

</body>

</html>
```

“Output”



11. Display random quotes on button click.

“randomQuotes.html”

```
<html>
```

```
<head>
```

```
<title>  
    Image change  
</title>  
  
<link rel="stylesheet" href="Button.css">  
  
<script language="jacvascript" type="text/javascript">  
  
    function changeQuote()  
  
    {  
  
        var number=Math.floor(Math.random() * 4);  
  
        var quoteSrc=["Anyone who has ever made anything of importance was  
disciplined." — Andrew Hendrixson",  
  
                    ""“Don’t spend time beating on a wall, hoping to transform it into a  
door.” — Coco Chanel",  
  
                    ""“Creativity is intelligence having fun.” — Albert Einstein",  
  
                    ""“Optimism is the one quality more associated with success and  
happiness than any other.” — Brian Tracy",  
  
                    " “Always keep your eyes open. Keep watching. Because whatever you  
see can inspire you.” — Grace Coddington"  
  
    };  
  
    document.getElementById("quotes").innerHTML=quoteSrc[number];  
  
}  
  
</script>  
  
</head>  
  
<body>  
  
    <p align="center" style="padding-top=100px;">  
  
        <div id="quotes">  
  
            "What you get by achieving your goals is not as important as what you  
become by achieving your goals." — Henry David Thoreau
```

```
</div>

<button onclick="changeQuote()" class="button">Click to change
Quote</button>

</p>

</body>

</html>
```

“Button.css”

```
body{

background-color:papayawhip;

font-family:Century;

text-align:center;

text-decoration:bold;

text-indent:left;

font-size:25px;

}
```

```
.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;
```

```
border-radius: 15px;  
box-shadow: 0 9px #999;  
}  
  
.button:hover {background-color: #3e8e41}
```

```
.button:active {  
background-color: #3e8e41;  
box-shadow: 0 5px #666;  
transform: translateY(4px);  
}
```

### “Output”





12. Add a new entry in combo box dynamically.

"Combox.html"

```
<html>
```

```
    <head>
```

```
        <title>
```

```
            Moving items to list box!
```

```
        </title>
```

```
        <link rel="stylesheet" href="comboBox.css">
```

```
    <script>
```

```
        function moveToSelected()
```

```
    {
```

```
        var flag=0;
```

```
        var items=document.getElementById("Fruits");
```

```
        var element = document.getElementById("newElement").value;
```

```
for(var i=0;i<items.length;i++)  
{  
  
if(!element.toLowerCase().localeCompare((items.options[i].text).toLowerCase())){  
  
    flag=1;  
  
    break;  
  
}  
  
}  
  
if(flag==0)  
{  
  
    var option = document.createElement("option");  
  
    option.text =element;  
  
    document.getElementById("Fruits").add(option);  
  
}  
  
}  
  
</script>  
  
</head>  
  
<body>  
  
<div align="center">  
  
<form>  
  
    <label>Enter fruit which you want to add:</label><input type="text"  
id="newElement">  
  
</form>  
  
    <button class="button" id="button1"  
onclick="moveToSelected()">Add</button><br><br>  
  
<form>  
  
    <select id="Fruits" class="myselect">
```

```
<option>Apple</option>
<option>Pear</option>
<option>Banana</option>
<option>Orange</option>
</select>
</form>
</div>
</body>
</html>
```

“combobox.css”

```
body{
background-color:papayawhip;
font-family:Century;
text-align:center;
text-decoration:bold;
text-indent:left;
font-size:25px;
}
```

```
.myselect {
display: block;
font-size: 16px;
font-family: sans-serif;
font-weight: 700;
color: #444;
line-height: 1.3;
```

```
padding: .6em 1.4em .5em .8em;  
width: 30%;  
max-width: 100%; /* useful when width is set to anything other than 100% */  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
-moz-appearance: none;  
-webkit-appearance: none;  
appearance: none;  
background-color: #fff;  
}
```

```
.myselect::ms-expand {  
display: none;  
}
```

```
.myselect:hover {  
border-color: #888;  
}  
/* Focus style */  
.button {  
display: inline-block;  
padding: 15px 25px;
```

```
font-size: 24px;  
cursor: pointer;  
text-align: center;  
text-decoration: none;  
outline: none;  
color: #fff;  
background-color: #4CAF50;  
border: none;  
border-radius: 15px;  
box-shadow: 0 9px #999;  
}
```

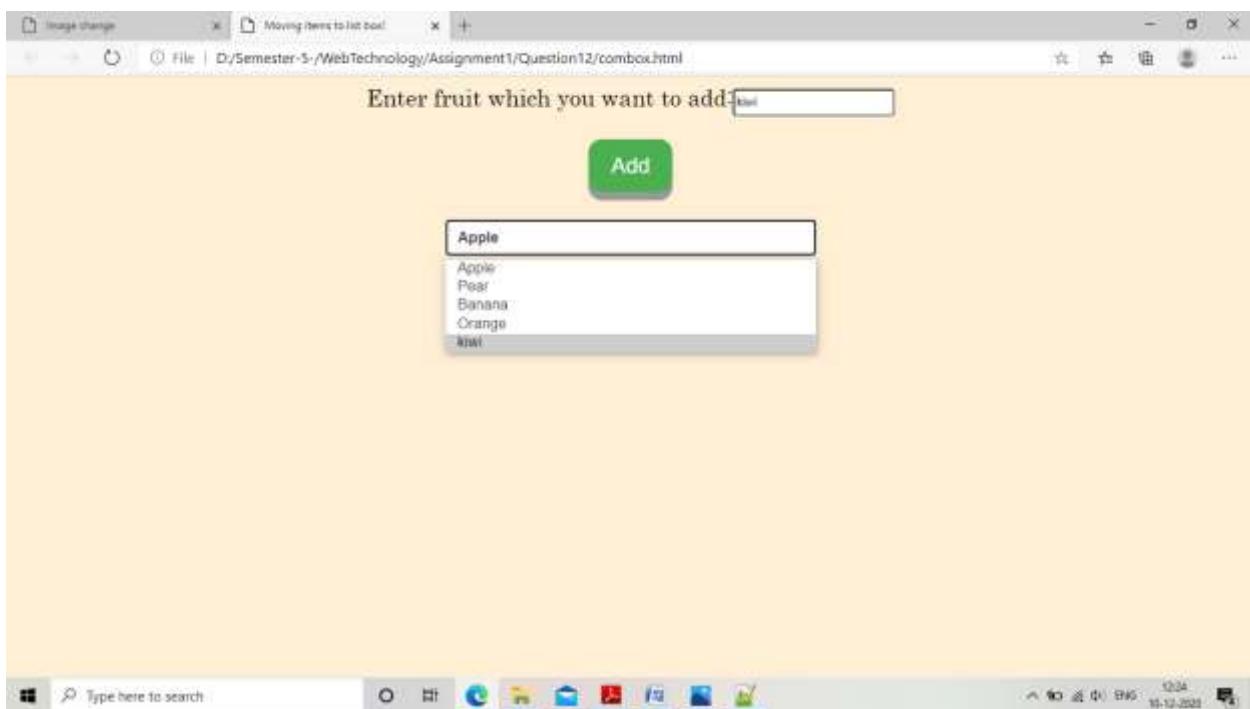
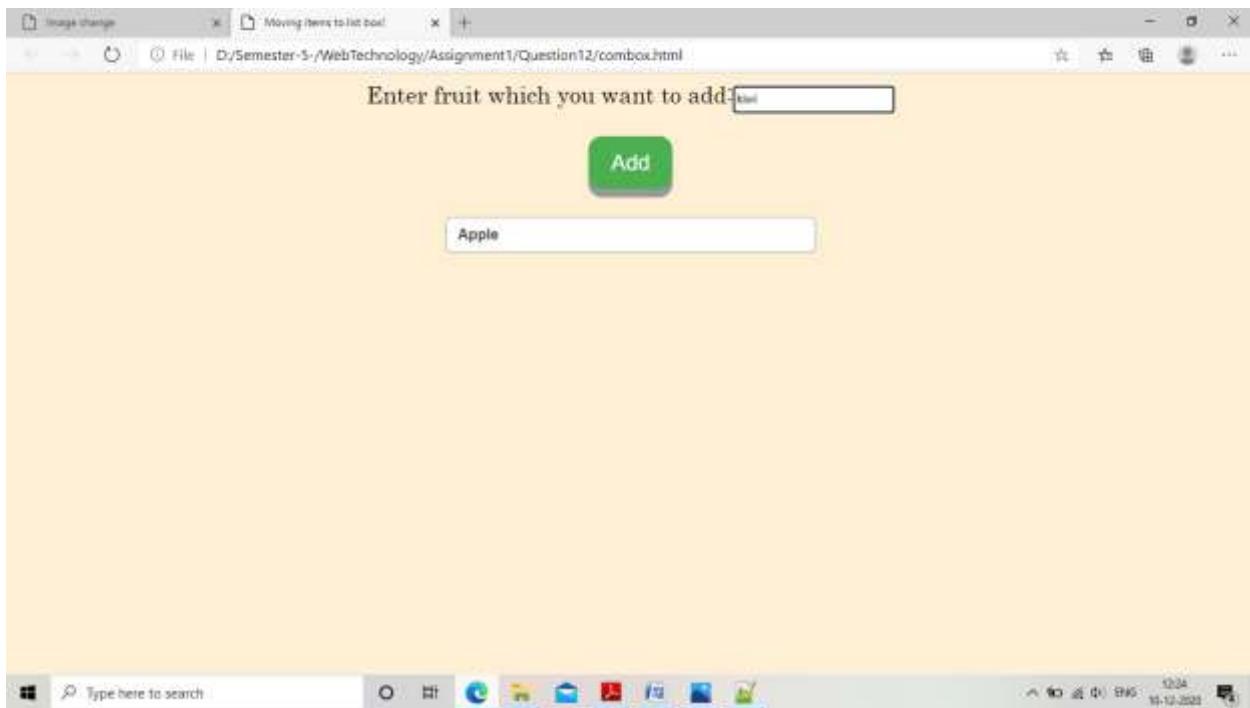
```
.button:hover {background-color: #3e8e41}
```

```
.button:active {  
background-color: #3e8e41;  
box-shadow: 0 5px #666;  
transform: translateY(4px);  
}  
  
input[type=text]{  
border-radius:4px;  
box-sizing:border-box;  
height:30px;  
}  
  
.select {
```

```
display: block;  
font-size: 16px;  
font-family: sans-serif;  
font-weight: 700;  
color: #444;  
line-height: 1.3;  
padding: .6em 1.4em .5em .8em;  
width: 100%;  
max-width: 100%;  
box-sizing: border-box;  
margin: 0;  
border: 1px solid #aaa;  
box-shadow: 0 1px 0 1px rgba(0,0,0,.04);  
border-radius: .5em;  
appearance: none;  
background-color: #fff;
```

```
}
```

“Output”



## Assignment-2

1. Create a JSP file which will display current date and time

“Question1.jsp”

```
<%@page import="java.util.Date" %>

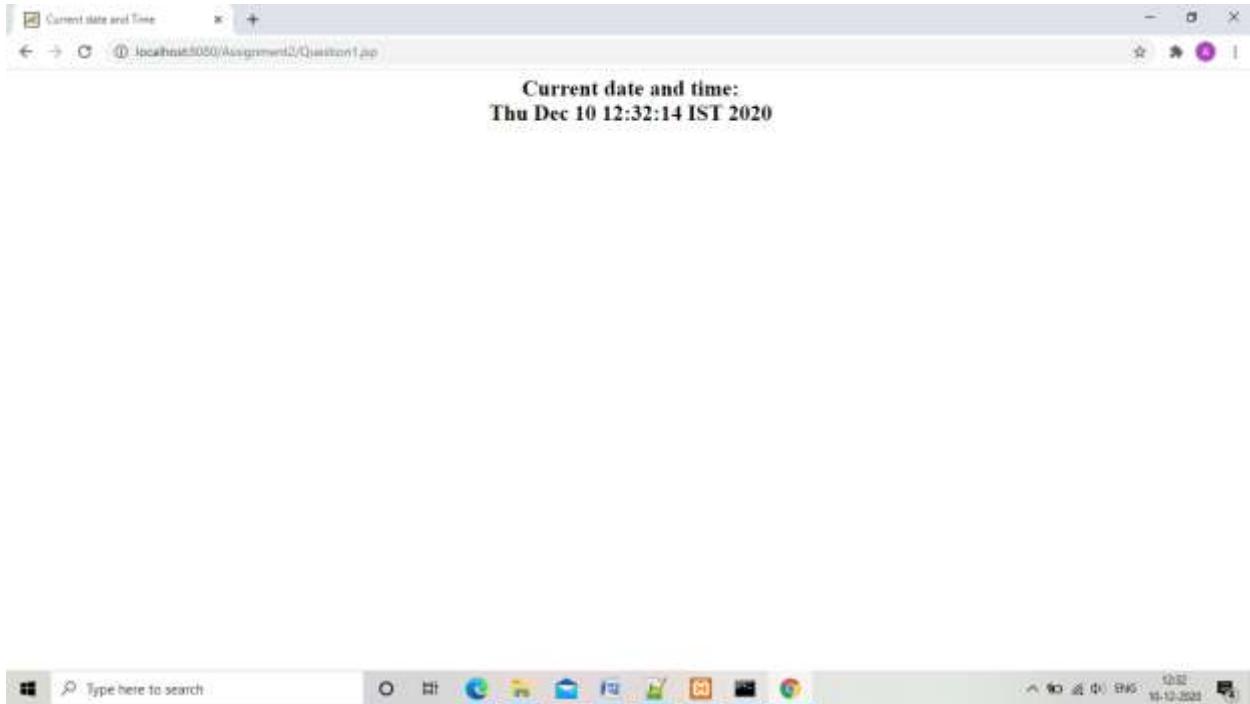
<html>

<head>
    <title>
        Current date and Time
    </title>
</head>

<body>
    <h2><p align="center">Current date and time:<br />
    <%
        Date today = new Date();
        out.println(today.toString());
    %>
</body>

<html>
```

“Output”



2. Create a JSP page for taking at least three parameters from user and display the details in same page

“Question2Form.html”

```
<html>
  <head>
    <title>
      User Input
    </title>
  </head>
  <body>
    <h2><p align="center">
      <form action="Question2.jsp" method="post">
        <label>Enter parameter1:</label>
        <input type="text" name="Parameter1" required><br /><br />
        <label>Enter parameter2:</label>
```

```

        <input type="text" name="Parameter2" required><br /><br />

        <label>Enter parameter2:</label>

        <input type="text" name="Parameter3" required><br /><br />

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

        <input type="reset" value="reset">

    </form>

</body>

<html>

    "Question2.jsp"

<html>

<head>

    <title>

        User paramters

    </title>

</head>

<body>

    <h2><p align="center">User parameters<br />

    <%
        out.println("Parameter1:"+request.getParameter("Parameter1"));

    %><br />

    <%
        out.println("Parameter2:"+request.getParameter("Parameter2"));

    %><br />

    <%
        out.println("Paramater3:"+request.getParameter("Parameter3"));

    %

```

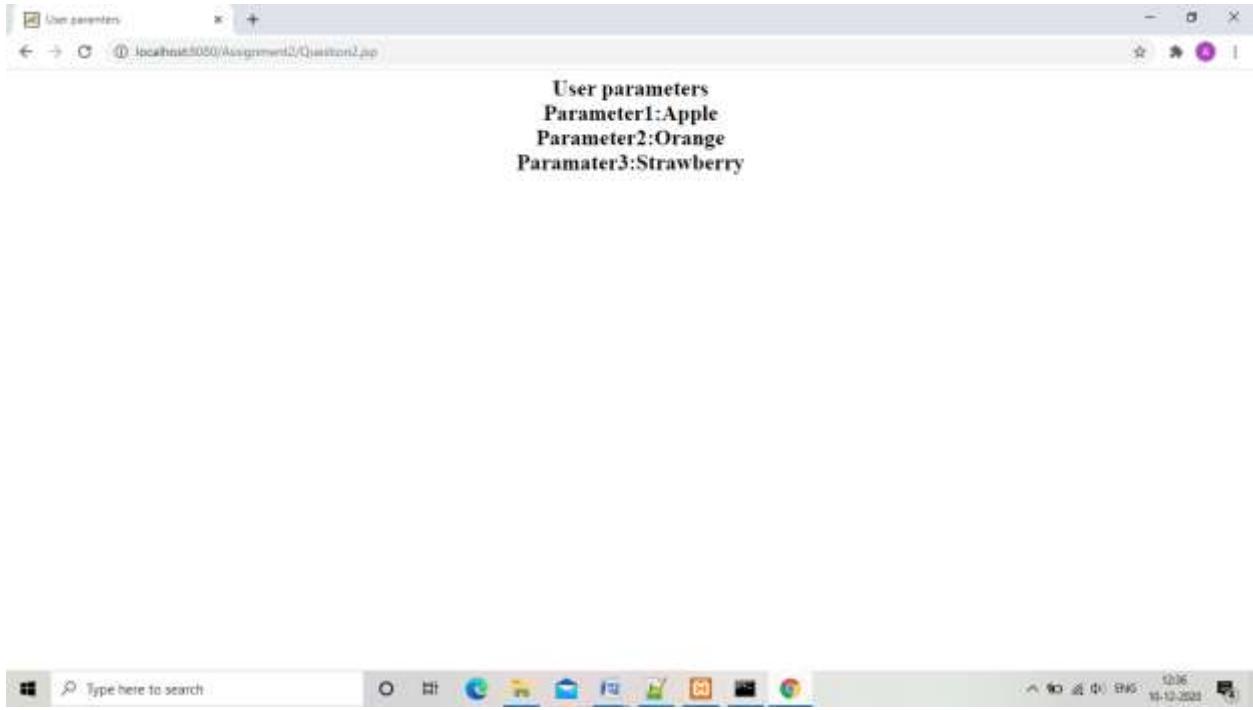
```
%><br />  
</p>  
</body>  
  
<html>
```

“Output”



A screenshot of a web browser window titled "User Input". The address bar shows the URL "localhost:8080/Assignment2/Question2Form.html". The page contains two text input fields: "Enter parameter1: " and "Enter parameter2: ". Below these inputs are two buttons: "submit" and "reset".





3.Create a jsp page that will take from user . If no parameter is supplied then print “hello world!”

“Question3Form.htm”

```
<html>
  <head>
    <title>
      User Input
    </title>
  </head>
  <body>
    <h2><p align="center">
      <form action="Question3.jsp" method="post">
        <label>Enter name:</label>
        <input type="text" name="name" ><br /><br />
        <input type="submit" value="submit">
        <input type="reset" value="reset">
```

```
</form>

</body>

<html>
    "Question3.jsp"

<html>

<head>
    <title>
        Welcome User
    </title>
</head>

<body>
    <h2><p align="center">Welcome User<br />
    <%
        String name = request.getParameter("name");
        if(name.equals(""))
            out.println("Hello world!");
        else
            out.println("Hello "+name);
    %>
    </p>
</body>

<html>
    "Output"
```

User Input

Enter name: Antra

Type here to search

Welcome User

Hello Antra

User Input

Enter name:



3. Create a JSP file which will take bgcolor as parameter from user and if none is passed then bgcolor will be set as RED

“Question4Form.html”

```
<html>
```

```
  <head>
```

```

<title>
    User Input
</title>

</head>

<body>

    <h2><p align="center">
        <form action="Question4.jsp" method="post">
            <label>Enter color:</label>
            <input type="text" name="color" ><br /><br />
            <input type="submit" value="submit">
            <input type="reset" value="reset">
        </form>
    </p>
</body>

<html>
    "Question4.jsp"

<html>

<head>
    <title>
        Change Background color
    </title>
</head>

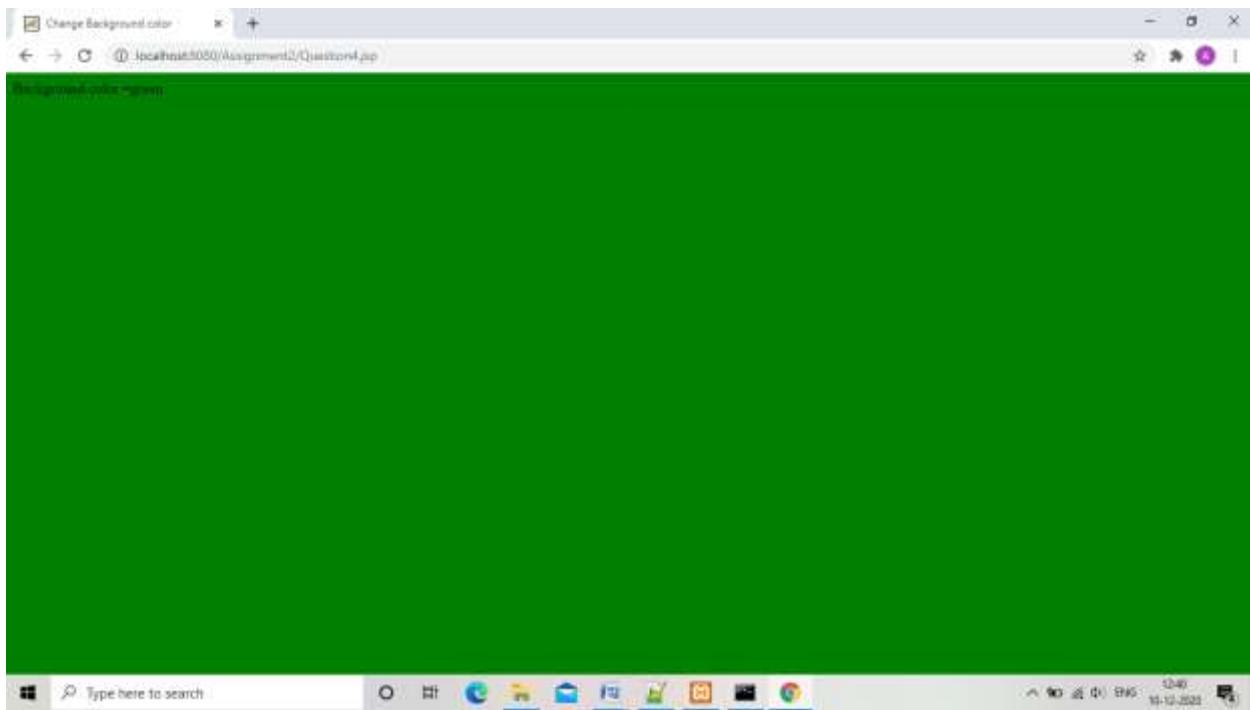
<%
    String color = request.getParameter("color");
    if(color.equals("")) {
        color="red";
    }
%>

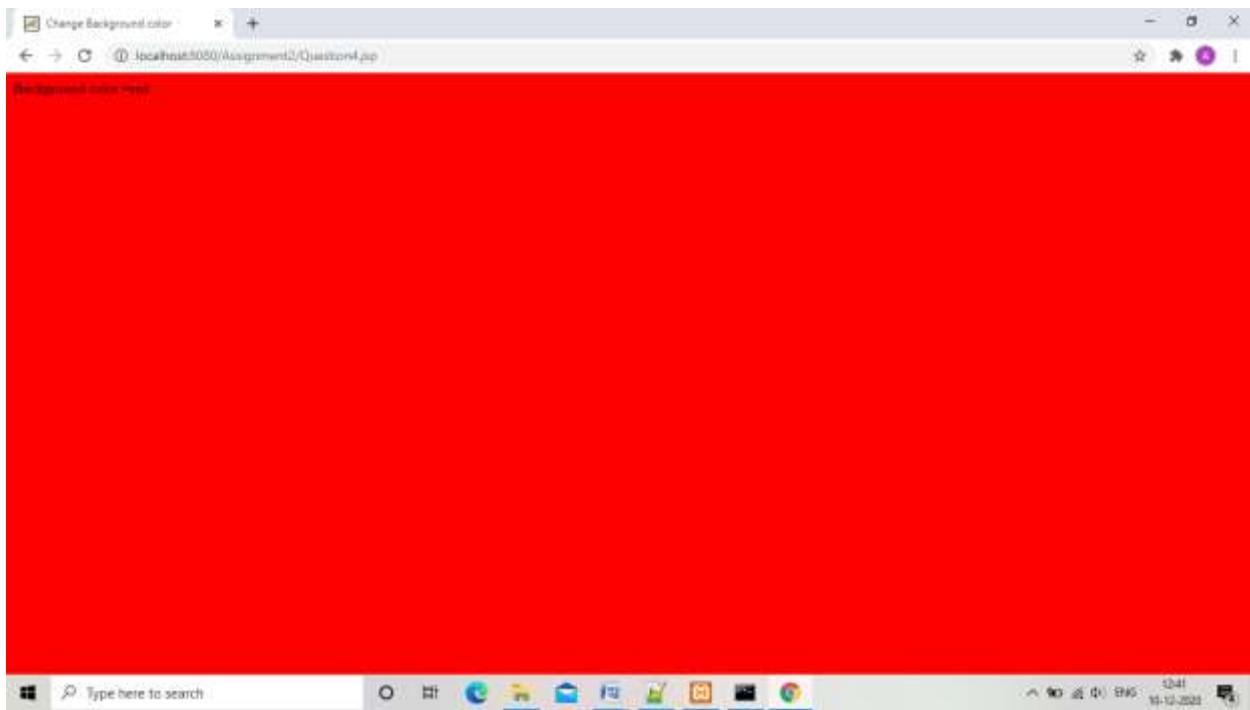
```

```
%>  
<body bgcolor="<%=\= color %>">  
<p>Background color =<% out.print(color);%>  
</body>  
<html>
```

“Output”







4. Write a jsp page which will print the number =r of times a current page has been accessed.

“Question5.jsp”

```
<html>
<head>
    <title>
        Hit count
    </title>
</head>
<body>
    <%
        Integer hit_counter = (Integer)application.getAttribute("hitCount");
        if(hit_counter==null || hit_counter==0){
            out.println("First time visit.Welcome!");
            hit_counter=1;
        }
    %>
```

```
    }

    else

    {

        hit_counter+=1;

    }

    application.setAttribute("hitCount",hit_counter);

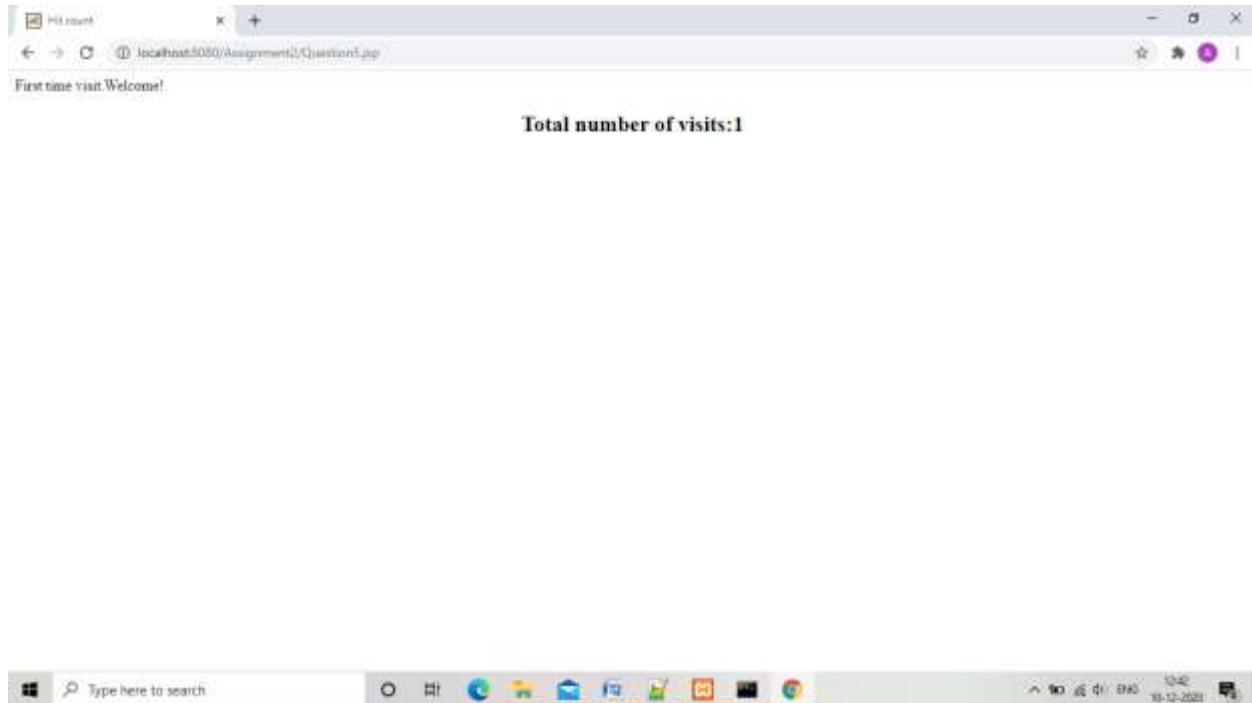
%>

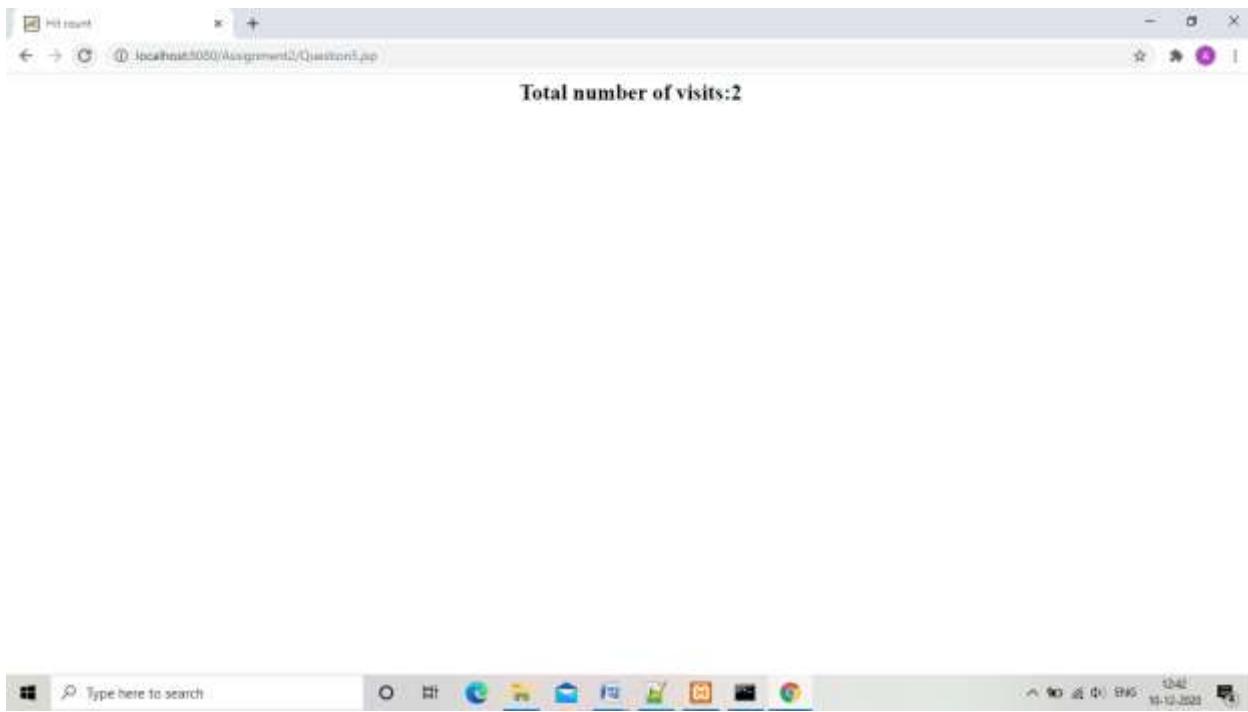
<h2 align="center">Total number of visits:<% out.print(hit_counter);%>

</body>

</html>
```

“Output”





6.Create a method that will print the addition of two numbers .

“Question6.jsp”

```
<html>
  <head>
    <title>
      User paramters
    </title>
  </head>
  <body>
    <h2><p align="center">Sum of 34 and 56 is:<br />
    <%!
      int sum(int number1,int number2)
      {
        return number1+number2;
      }
    
```

```
%>  
<%  
    out.println(sum(34,56));  
%><br />  
</p>  
</body>  
<html>
```

“Output”



7. Allow the user to enter the name in the textbox and when he click on submit button, print the user name and send the data using GET method. If user click on submit without UserName then ask the user to enter it

“Question7Form.html”

```
<html>  
<head>  
<title>
```

## User Input

```
</title>

</head>

<body>

<h2><p align="center">

<form action="Question7.jsp" method="get">

<label>Enter name:</label>

<input type="text" name="name" ><br /><br />

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

<input type="reset" value="reset">

</form>

</body>

<html>

    "Question7.jsp"

<html>

<head>

    <title>

        Welcome User

    </title>

</head>

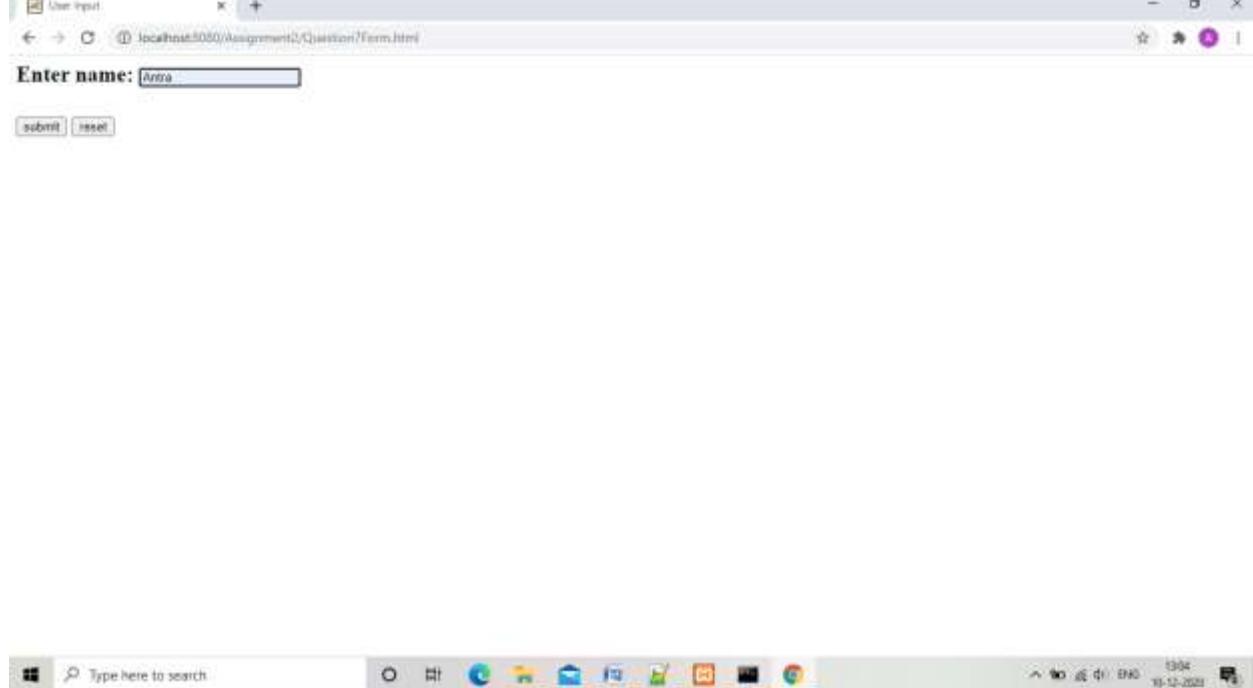
<body>

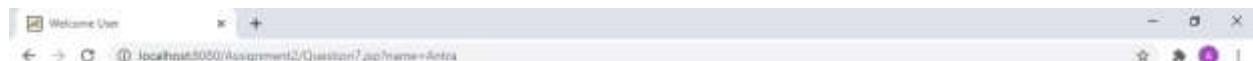
    <h2><p align="center">Welcome User<br />

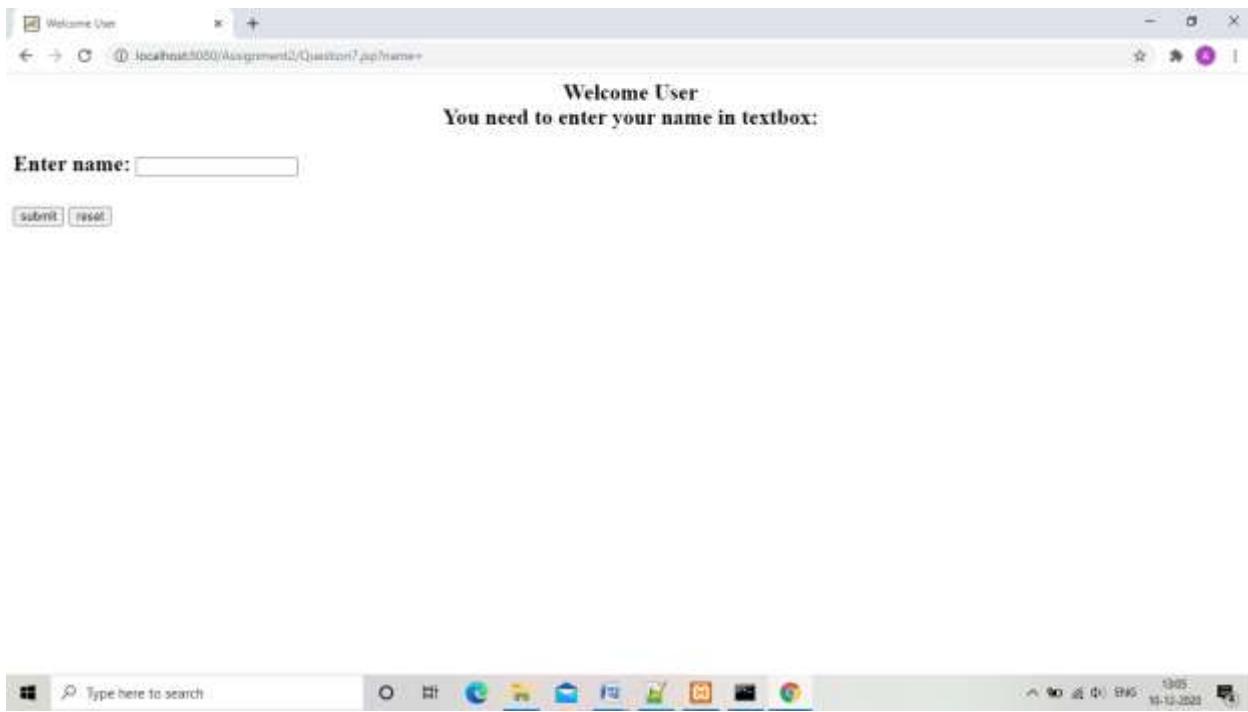
    <%
        String name = request.getParameter("name");
        if(name.equals(""))
    
```

```
{  
    out.println("You need to enter your name in textbox:");  
%>  
<%@ include file="Question7Form.html" %>  
  
<%  
}  
else  
    out.println("Hello "+name);  
%>  
</p>  
</body>  
<html>
```

“Output”







8. Execute the above program with “post”

“Question8Form.html”

```
<html>
  <head>
    <title>
      User Input
    </title>
  </head>
  <body>
    <h2><p align="center">
      <form action="Question8.jsp" method="post">
        <label>Enter name:</label>
        <input type="text" name="name" ><br /><br />
        <input type="submit" value="submit">
    
```

```
<input type="reset" value="reset">  
</form>  
</body>  
<html>
```

“Question8.jsp”

```
<html>  
  <head>  
    <title>  
      Welcome User  
    </title>  
  </head>  
  <body>  
    <h2><p align="center">Welcome User<br />  
    <%  
      String name = request.getParameter("name");  
      if(name.equals(""))  
      {  
        out.println("You need to enter your name in textbox:");  
      }  
    %>  
    <%@ include file="Question8Form.html" %>
```

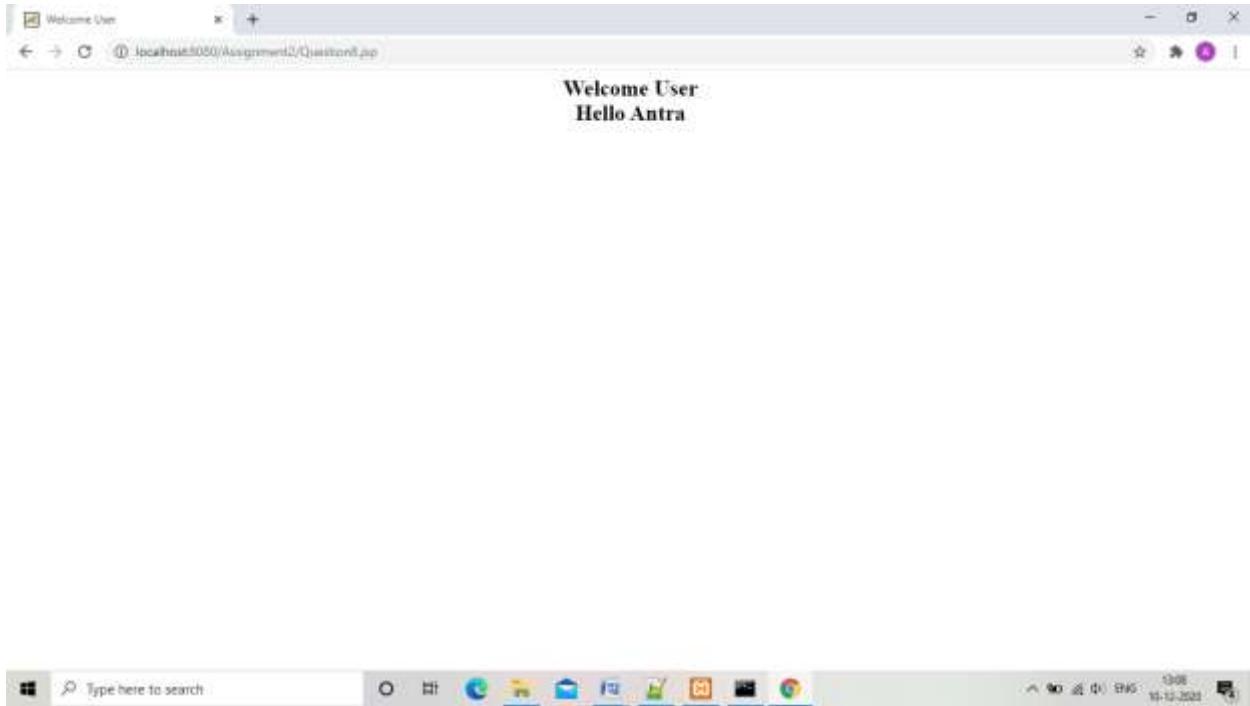
```
<%  
}  
else  
  out.println("Hello "+name);
```

```
%>  
</p>  
</body>
```

```
<html>
```

“Output”





9. Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, a background color should be selected at random.

“Question9Form.html”

```
<html>
<head>
    <title>
        User Input
    </title>
</head>
<body>
    <h2><p align="center">
        <form action="Question9.jsp" method="post">
            <label>Enter color:</label>
            <input type="text" name="color" ><br /><br />
            <input type="submit" value="submit">
        </form>
    </p>
</body>
```

```

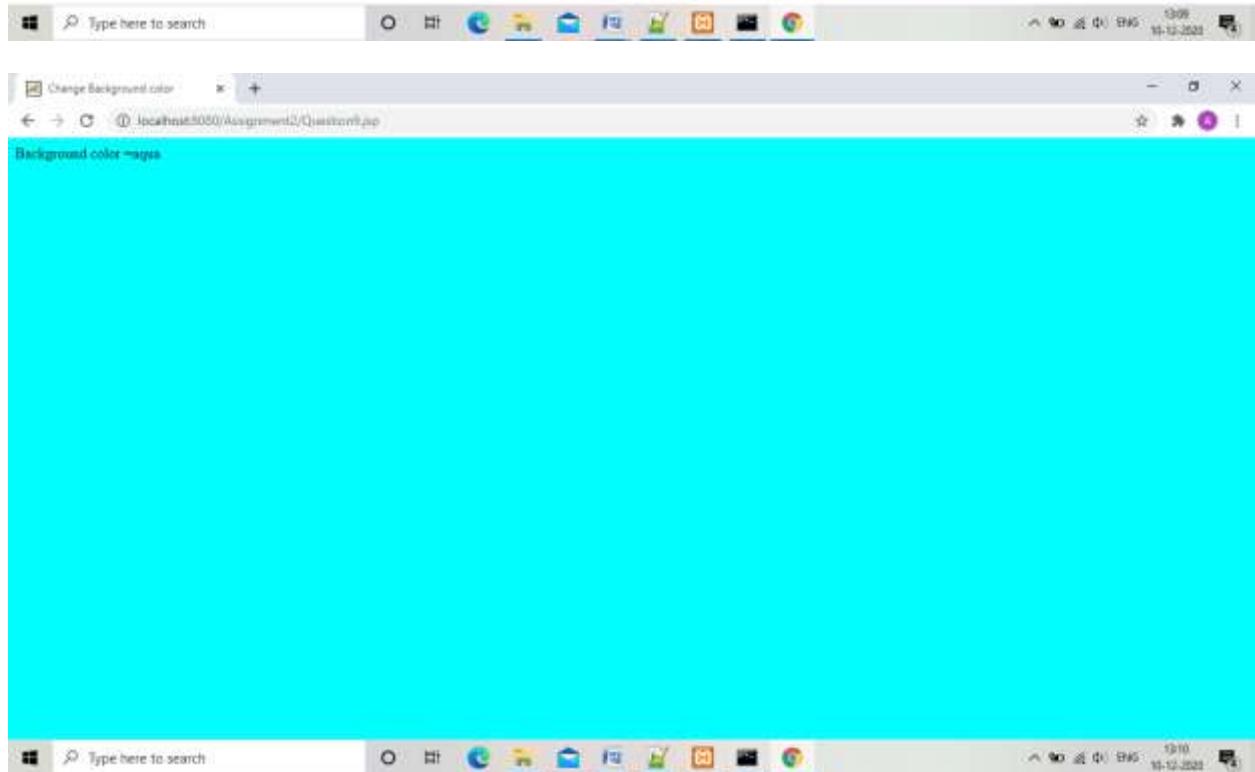
<input type="reset" value="reset">
</form>
</body>
<html>
    "Question9.jsp"
<%@ page import="java.util.Random" %>
<html>
    <head>
        <title>
            Change Background color
        </title>
    </head>
    <%
        String color = request.getParameter("color");
        if(color.equals(""))
        {
            Random random = new Random();
            String[] colors={"AliceBlue","light salmon","AntiqueWhite","Beige"};
            int i = random.nextInt(4);
            color=colors[i];
        }
    %>
    <body bgcolor="<%= color %>">
        <p>Background color =<% out.print(color);%>

```

```
</body>
```

```
<html>
```

“Output”



10. Write a jsp page to greet user randomly.

“Question10.jsp”

```
<%@ page import="java.util.Date" %>

<html>

<head>
    <title>
        Welcome User
    </title>
</head>

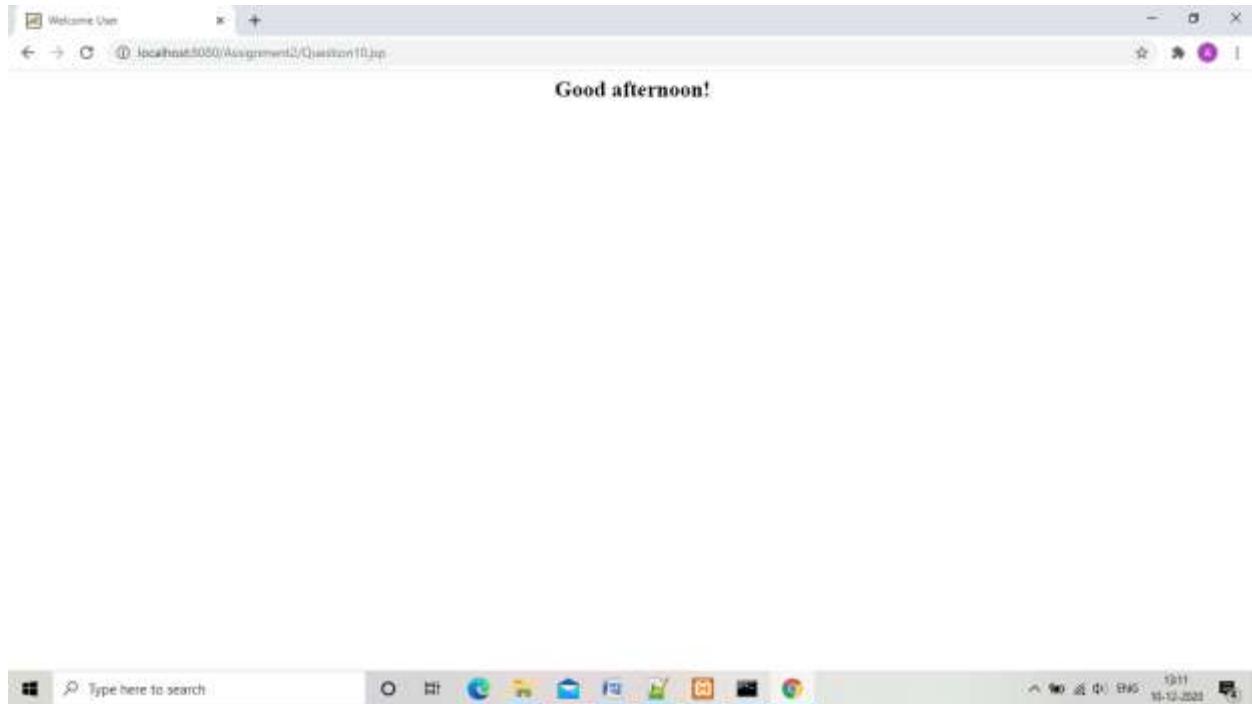
<body>
    <h2><p align="center">
        <%
            Date today = new Date();
            int hour = today.getHours();
            if(hour<=12){
                out.print("Good morning!");
            }
            else if(hour>12 && hour<=17){
                out.print("Good afternoon!");
            }
            else if(hour>17 && hour<=23){
                out.print("Good evening!");
            }
        %>
    </p>

```

```
</body>
```

```
<html>
```

“Output”



11. Create an application which will show any random number between 1 and 10 (Make proper use of import tag)

“Question11.jsp”

```
<%@ page import="java.util.Random" %>
```

```
<html>
```

```
<head>
```

```
<title>
```

Change Background color

```
</title>
```

```
</head>
```

```
<body>
```

```
<%
    Random random = new Random();
    int i = random.nextInt(500);
%>
<p>Random number generated is : =<% out.print(i);%>
</body>
<html>
```

“Output”



12. Create a JSP file which will take number from the user in textbox and divide the number by 5 .If any error occurs display a proper error page.

“Question12Form.html”

```
<html>
<head>
<title>
User Input
```

```

        </title>

    </head>

    <body>

        <h2><p align="center">

            <form action="Question12.jsp" method="post">

                <label>Enter Number:</label>

                <input type="text" name="number" ><br /><br />

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

                <input type="reset" value="reset">

            </form>

        </body>

<html>

        "Question12.jsp"

<%@ page errorPage="ErrorPage12.jsp" %>

<html>

    <head>

        <title>

            Divide by 5

        </title>

    </head>

    <body>

        <%
            int number=Integer.parseInt((String)request.getParameter("number"));

            out.println(number/5);

        %>

```

```
<p>  
</body>  
  
<html>  
  
    “ErrorPage12.jsp”  
  
<%@ page isErrorPage="true" %>  
  
<html>  
  
    <head>  
  
        <title>  
  
            Divide by 5  
  
        </title>  
  
    </head>  
  
    <body>  
  
        <h2>Sorry there was an error!  
  
        <p>  
  
    </body>  
  
<html>  
  
    “Output”
```

User Input

Enter Number:

Type here to search

Divide by 5

← → ⌂ ⓘ 15:15 10-12-2021

3

User Input

Enter Number:

submit reset

Type here to search

Divide by 5

Sorry there was an error!



13. Generate 10 random number in the range of 1-10 and print it in the first column of table.If the number generated is even the second column should display-Even Number else Odd number.

“Question13.jsp”

```
<%@ page contentType="application/vnd.ms-excel" import="java.util.Random" %>
```

```
<%!
```

```
Random random = new Random();

boolean check(int number){

    if(number % 2 == 0)

        return true;

    else

        return false;

}

%>

<%

for(int i =0;i<10;i++){

    int number = random.nextInt(10);

    if(check(number)){

        out.println(number+"\t"+"Even Number");

    }

    else{

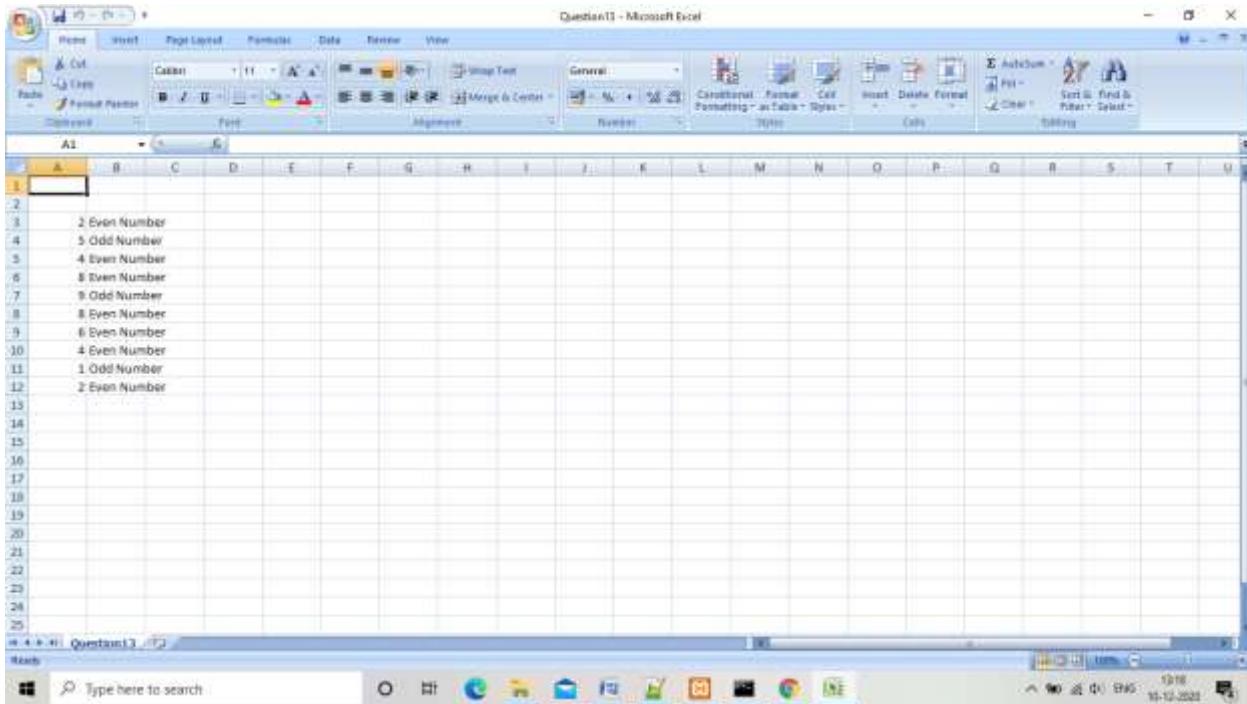
        out.println(number+"\t"+"Odd Number");

    }

}

%>

“Output”
```



14. Create an application that displays the count of number of times any of the page has been accessed by the user. Make atleast 4 different JSP pages.If the user enter the URL of any jsp page for the first time It shows the message-“Welcome You have visited this page for the first time” and next time when user enter the same URL it show the message showing no.of times this JSP page has been visited.

“Question14HomePage.jsp”

```
<html>
<head>
    <title>
        Home Page
    </title>
</head>
<body>
    <%
        Integer hit_counter = (Integer)application.getAttribute("HomePage");
        if(hit_counter==null || hit_counter==0){
            out.println("First time visit.Welcome!");
        }
    %>
```

```

        hit_counter=1;

    }

else

{

    hit_counter+=1;

}

application.setAttribute("HomePage",hit_counter);

%>

<center><h2>Number of visits:<% out.println(hit_counter); %>

<a href="Question14Page2.jsp">Go to Page2</a>

</body>

</html>

"Question14Page2.jsp"

<html>

<head>

    <title>

        Home Page

    </title>

</head>

<body>

<%

Integer hit_counter = (Integer)application.getAttribute("Page2");

if(hit_counter==null || hit_counter==0){

    out.println("First time visit.Welcome!");

    hit_counter=1;

```

```

        }

        else

        {

            hit_counter+=1;

        }

        application.setAttribute("Page2",hit_counter);

    %>

<center><h2>Number of visits:<% out.println(hit_counter); %>

<a href="Question14Page3.jsp">Go to Page3</a>

</body>

</html>

        "Question14Page3.jsp"

<html>

<head>

    <title>

        Home Page

    </title>

</head>

<body>

<%

        Integer hit_counter = (Integer)application.getAttribute("Page3");

        if(hit_counter==null || hit_counter==0){

            out.println("First time visit.Welcome!");

            hit_counter=1;

        }


```

```

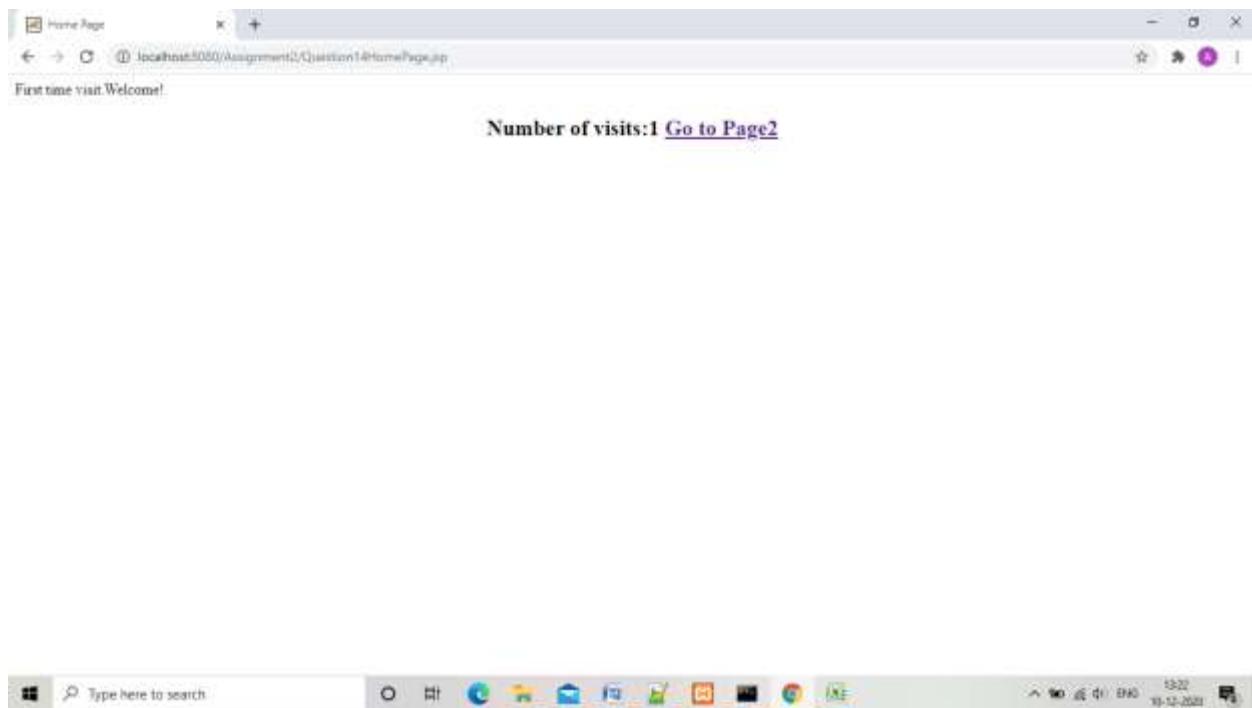
        else
        {
            hit_counter+=1;
        }
        application.setAttribute("Page3",hit_counter);
    %>
<center><h2>Number of visits:<% out.println(hit_counter); %>
<a href="Question14Page4.jsp">Go to Page4</a>
</body>
</html>
"Question14Page4.jsp"

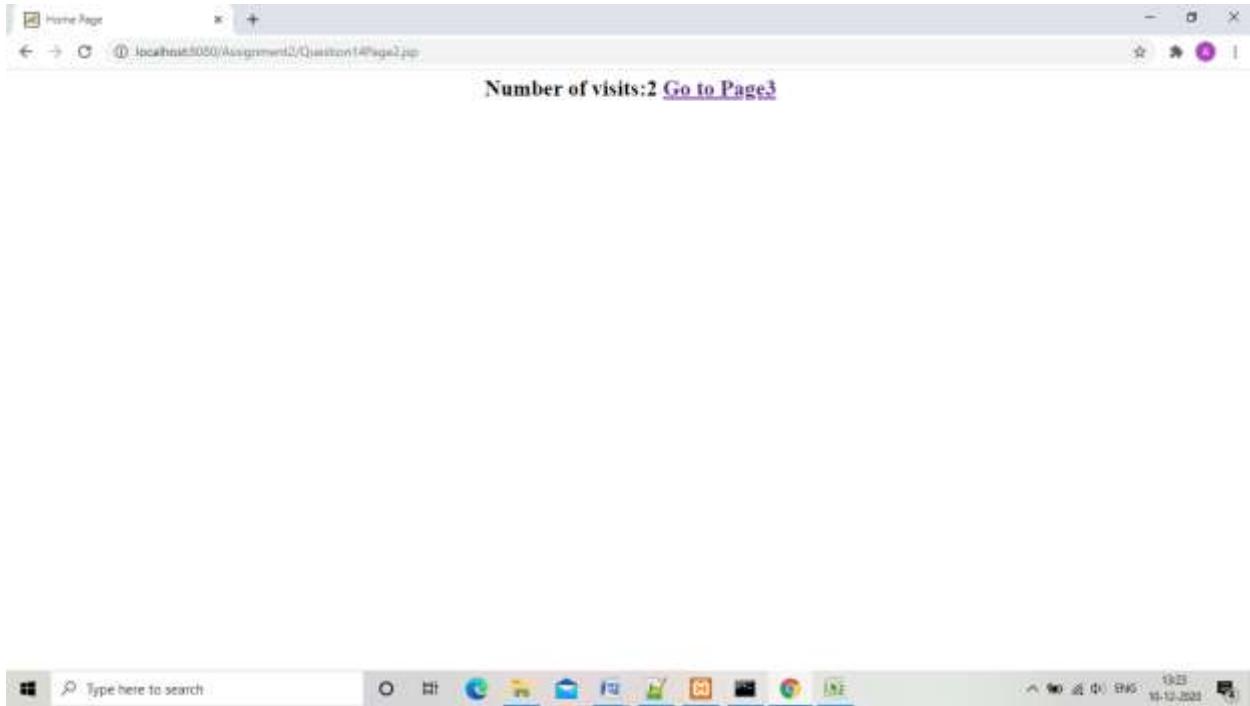
<html>
<head>
<title>
    Home Page
</title>
</head>
<body>
<%
    Integer hit_counter = (Integer)application.getAttribute("Page4");
    if(hit_counter==null || hit_counter==0){
        out.println("First time visit.Welcome!");
        hit_counter=1;
    }
    else

```

```
{  
    hit_counter+=1;  
}  
  
application.setAttribute("Page4",hit_counter);  
  
%>  
  
<center><h2>Number of visits:<% out.println(hit_counter); %>  
  
</body>  
  
</html>
```

“Output”





15. Design an application that accepts distance and the time taken to cover the distance of city A to B ,B to C and C to D from user and finds the speed. If the error occurs, a page Error.jsp should be displayed, otherwise it should call a page showing the full description of journey- Time speed and distance . the output should be in excel format.

“Question15Form.html”

```
<html>
  <head>
    <title>
      User Input
    </title>
  </head>
  <body>
    <form action="Question15.jsp" method="post">
      <label>Enter distance from city A to B:</label>
      <input type="text" name="distance_a_b"><br /><br />
      <label>Enter time to travel A to B:</label>
```

```

<input type="text" name="time_a_b"><br /><br />
<br /><br /><br />

<label>Enter distance from city B to C:</label>
<input type="text" name="distance_b_c"><br /><br />
<label>Enter time to travel B to C:</label>
<input type="text" name="time_b_c"><br /><br />
<br /><br /><br />

<label>Enter distance from city C to D:</label>
<input type="text" name="distance_c_d"><br /><br />
<label>Enter time to travel C to D:</label>
<input type="text" name="time_c_d"><br /><br />
<br /><br /><br />

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

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

"Question15.jsp"

<%@ page errorPage="ErrorPage15.jsp" contentType="application/vnd.ms-excel" %>
<html>
<head>
<title>
    Journey
</title>
</head>

```

```

<body>

<table>

<tr>

    <th><%out.println("Distance");%></th>
    <th><%out.println("Time");%></th>
    <th><%out.println("Speed");%></th>

</tr>

<%
    int distance=Integer.parseInt((String)request.getParameter("distance_a_b"));

    int time=Integer.parseInt((String)request.getParameter("time_a_b"));

    int speed=distance/time;

%>

<tr>

    <td><%out.print(distance);%></td>
    <td><%out.print(time);%></td>
    <td><%out.print(speed);%></td>

</tr>

<%
    distance=Integer.parseInt((String)request.getParameter("distance_b_c"));

    time=Integer.parseInt((String)request.getParameter("time_b_c"));

    speed=distance/time;

%>

<tr>

    <td><%out.print(distance);%></td>
    <td><%out.print(time);%></td>

```

```

<td><%out.print(speed);%></td>
</tr>
<%
    distance=Integer.parseInt((String)request.getParameter("distance_c_d"));
    time=Integer.parseInt((String)request.getParameter("time_c_d"));
    speed=distance/time;
%>
<tr>
    <td><%out.print(distance);%></td>
    <td><%out.print(time);%></td>
    <td><%out.print(speed);%></td>
</tr>
</table>
</body>
<html>
    “ErrorPage15.jsp”
<%@ page isErrorPage="true" %>
<html>
    <head>
        <title>
            Journey
        </title>
    </head>
    <body>
        <h2>Sorry there was an error!

```

<p>

</body>

<html>

“Output”

The screenshot shows a Microsoft Edge browser window with the title bar "User Input". The address bar displays the URL "localhost:8080/Assignment2/Question1Sform.html". The page content is a form for calculating travel times between four cities (A, B, C, D). The form fields are as follows:

- Enter distance from city A to B:
- Enter time to travel A to B:
- Enter distance from city B to C:
- Enter time to travel B to C:
- Enter distance from city C to D:
- Enter time to travel C to D:

At the bottom of the form are two buttons: "submit" and "reset".

Below the browser window, the Windows taskbar is visible, showing the Start button, a search bar with the placeholder "Type here to search", and icons for various pinned apps like File Explorer, Mail, and Photos. The system tray shows the date and time as "10-12-2021 10:25".

Distance	Time	Speed
25	5	5
30	5	6
35	5	7

### “Assignment-3”

1. Create a Jsp page which provides student detail ,store the detail in student bean and show a Jsp page which display all the details.

### “StudentData.java”

```
package student;
```

```
public class StudentData{
    private String fname,lname,address;
    int age;
    public void setFname(String fname){
        this.fname=fname;
    }
    public String getFname(){
```

```
        return this.fname;  
    }  
  
    public void setLname(String lname){  
        this.lname=lname;  
    }  
  
    public String getLname(){  
        return this.lname;  
    }  
  
    public void setAddress(String address){  
        this.address=address;  
    }  
  
    public String getAddress(){  
        return this.address;  
    }  
  
    public void setAge(int age){  
        this.age=age;  
    }  
  
    public int getAge(){  
        return this.age;  
    }  
}
```

“StudentForm.html”

```
<html>  
  <head>  
    <title>
```

User Input

```
</title>

</head>

<body>

<form action="Question1.jsp" method="post">

    <label>Enter first name:</label>

    <input type="text" name="fname"><br /><br />

    <label>Enter last name:</label>

    <input type="text" name="lname"><br /><br />

    <label>Enter address</label>

    <input type="text" name="address"><br /><br />

    <label>Enter age:</label>

    <input type="text" name="age"><br /><br />

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

    <input type="reset" value="reset">

</form>

</body>

</html>
```

“Question1.jsp”

```
<html>

<head>

    <title>

        Student details

    </title>
```

```
</head>

<body>

<center><h2>

    <jsp:useBean id="student" class="student.StudentData" />

    <jsp:setProperty name="student" property="*" />

    <b><u>Student details:</u></b><br /><br />

    First name:<jsp:getProperty name="student" property="fname" /><br />

    Last name:<jsp:getProperty name="student" property="lname" /><br />

    Address:<jsp:getProperty name="student" property="address" /><br />

    Age:<jsp:getProperty name="student" property="age" /><br />

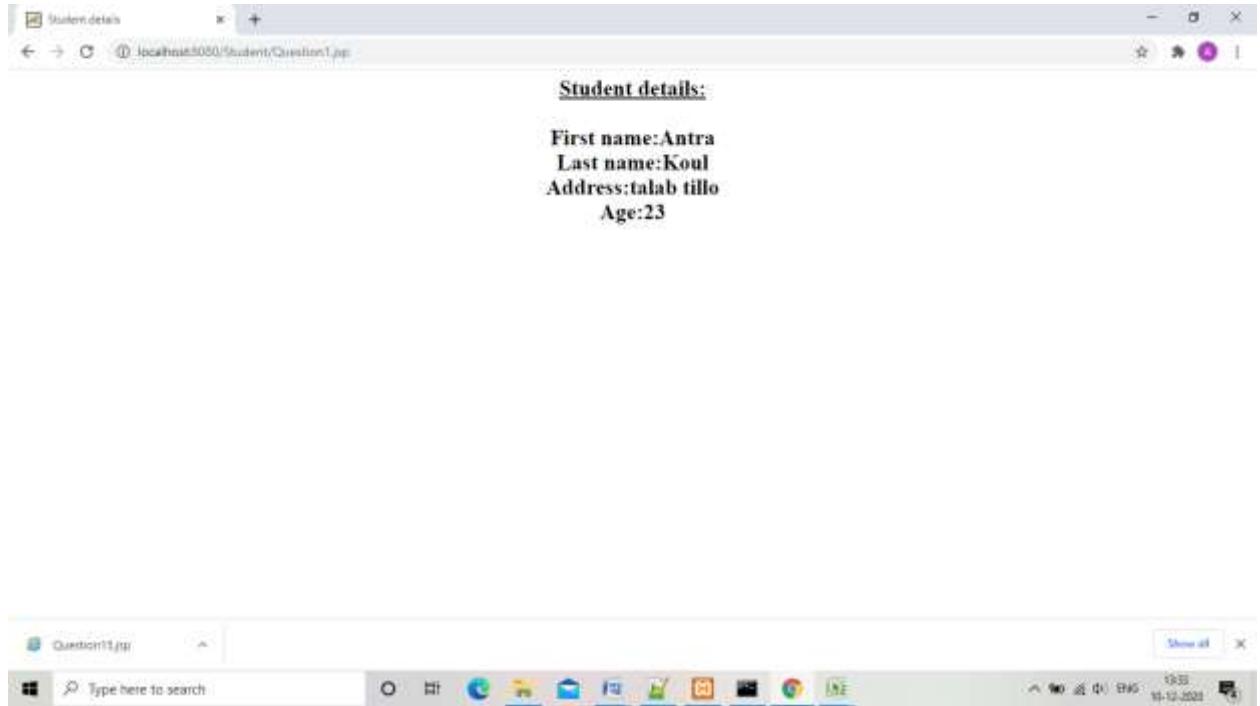
</center>

</body>

</html>
```

## “Output”





2. Create a color bean which has attributes userName background and foreground color. Create a Jsp page which ask the user –name and color choices and display the page with user specified colors with a message- Welcome <username>.

"UserForm.html"

```
<html>
    <head>
        <title>
            User Input
        </title>
    </head>
    <body>
        <h2><p align="center">
            <form action="UserPage.jsp" method="post">
                <label>Enter username:</label>
                <input type="text" name="username" required><br /><br />
                <label>Enter background color:</label>
                <input type="text" name="backgroundcolor" required><br />
            /><br />
                <label>Enter foreground color:</label>
                <input type="text" name="foregroundcolor" required><br /><br />
            </form>
        </p>
    </body>
<html>
```

“UserPage.jsp”

```
<html>
    <head>
        <title>
            User welcome
        </title>
    </head>
    <body>
        <jsp:useBean id="user1" class="user.UserBean" />
        <jsp:setProperty property="*" name="user1" />
        <body bgcolor=<jsp:getProperty name='user1' property='backgroundcolor' />" text="forecolor:<jsp:getProperty name='user1' property='foregroundcolor' />"
            Welcome!<jsp:getProperty name='user1' property='username' />
        </body>
    </html>
```

“UserBean.java”

```
package user;

public class UserBean{
    private String backgroundcolor,foregroundcolor,username;
    public void setUsername(String username){
        this.username=username;
    }
    public String getUsername(){
        return this.username;
    }
    public void setBackgroundcolor(String backgroundcolor){
        this.backgroundcolor=backgroundcolor;
    }
    public String getBackgroundcolor(){
        return this.backgroundcolor;
    }
    public void setForegroundcolor(String foregroundcolor){
        this.foregroundcolor=foregroundcolor;
    }
    public String getForegroundcolor(){
        return this.foregroundcolor;
    }
}
```

“Output”

User Input

Enter username: Antra

Enter background color: beige

Enter foreground color: red

submit reset

Welcome!Antra

3. Create a Jsp page ask user to select Product from drop down menu , enter quantity and show a page bill.jsp which shows the bill in table format. If the bill amount is less than 200 the page should have background color white else green.

“HomePage.html”

```
<html>
  <head>
```

```

<title>
    Home page
</title>
</head>
<body>
    <center><h2><i><u>Welcome ! </u></i><br /><br />
    <form method="post" action="Bill.jsp">
        <label>Select item:</label>
        <select name="product" id="product">
            <option value="notebook">Notebook</option>
            <option value="pen">Pen</option>
            <option value="colorpencils">Color pencils</option>
        </select>
        <label>Enter quantity:</label>
        <input type="text" name="quantity"><br /><br />
    <center>
        <input type="submit" value="submit">
        <input type="reset" value="reset">
    </form>
</body>
<html>

```

“Bill.jsp”

```

<html>
    <head>
        <title>
            Bill page
        </title>
    </head>
    <%
        String product = (String)request.getParameter("product");
        Integer quantity = Integer.parseInt(request.getParameter("quantity"));
        Integer price=0;
        if(product.equals("notebook")){
            price = quantity * 50;
        }
        else if(product.equals("pen")){
            price = quantity * 30;
        }
        else if(product.equals("colorpencils")){
            price = quantity * 150;
        }
        String color;
        if(price<=200)
        {
            color="white";
        }
        else{

```

```

        color="green";
    }
    %>
<body bgcolor="<% color%>"><h2>
<center><i><u>Your bill:</u></i><br />
<table width="50%" height="50%">
<tr>
    <th>Product name</th>
    <th>Quantity:</th>
    <th>Total</th>
</tr>
<tr>
    <th><%out.print(product);%></th>
    <th><%out.print(quantity);%></th>
    <th><%out.print(price);%></th>
</tr>
</table>
</body>
<html>
```

“Shop.css”

```

table {
    border-collapse: collapse;
}
td {
    text-align: center;
}
body{
    font-family:Comic Sans MS;
}
input[type=button], input[type=submit], input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
}
```

“Output”

A screenshot of a web browser window titled "Home page". The address bar shows the URL "localhost:5050/ShoppingCart/Homepage.html". The page content includes a welcome message "Welcome!" and a form with fields "Select item: Pen" and "Enter quantity: ". Below the input field are two buttons: "submit" and "reset".

A screenshot of a web browser window titled "Question15.jsp". The address bar shows the URL "localhost:5050/ShoppingCart/Bill.jsp". The page content includes a heading "Your bill:" and a table with three columns: "Product name", "Quantity:", and "Total". A single row is present in the table with the values "pen", "4", and "120".



#### "Assignment 4"

1. (a) Create an application for bank using JSTL. Customer enters his accountid , server validate the customer and display his account balance – Welcome customer name> your account balance is <balance> If the balance is below min balance then display appropriate message and change the background color.

## “Question1.jsp”

```

<c:set var="flag" value="1" />

<h2 >Customer Information </h2>

Welcome <c:out value="${Name}" /> !<br />

Your balance is <c:out value="${table.Balance}" /> <br />

<c:set var="balance" value="${table.Balance}" />

<c:if test="${balance < 2000}">

    Your balance is less than minimum balance required
    .Kindly deposit money otherwise your account will get deactivated!

    <script>
        document.body.style.backgroundColor="red";
    </script>

    </c:if>

</c:if>

</c:forEach>

<c:if test="${flag != '1'}">

    <c:out value="Sorry! wrong id" /></i>
    <c:import url="Question1Form.html" />

    </c:if>

    </h3>

    </center>

</body>

</html>

```

“Question1Form.html”

```
<html>
```

```
<head>
    <title>Customer Form</title>
</head>
<body>
    <center>
        <h2><u>Customer Form</u><br /><br />
        <form method="post" action="Question1.jsp">
            <label>Enter customer id:</label>
            <input type="text" name="customer_id"><br /><br />
            <input type="submit" value="submit">
            <input type="reset" value="reset">
        </form>
    </center>
</body>
</html>
```

“Output”

localhost / 127.0.0.1 / MySQL / accounts

phpMyAdmin

Recent: Favorites

New

information\_schema

jet

  + u\_hm

  + accounts

  + course

  + teferson

  + guestbook

  + product

  + profile

  + student

  + subject

  + user

mysql

performance\_schema

phpmyadmin

test

localhost / 127.0.0.1 / MySQL / jet / accounts

Show rows 0 - 3 (4 total, Query took 0.5444 seconds)

SELECT \* FROM `accounts`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

ID	Name	Balance
101	Asma kass	10000.85
102	Asma Nayant	1000
103	Khushaf Panchal	23000.89
104	Kriti Mistry	8800

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: Name

Options [Edit] [Copy] [Delete] 101 Asma kass 10000.85

Options [Edit] [Copy] [Delete] 102 Asma Nayant 1000

Options [Edit] [Copy] [Delete] 103 Khushaf Panchal 23000.89

Options [Edit] [Copy] [Delete] 104 Kriti Mistry 8800

Check all With selected Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: Name

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / MySQL / accounts

Customer Form

Enter customer id:

Submit Reset

localhost / 127.0.0.1 / MySQL / accounts

Customer Form

Enter customer id:

Submit Reset

localhost / 127.0.0.1 / MySQL / accounts

Customer Form

Enter customer id:

Submit Reset

localhost / 127.0.0.1 / MySQL / accounts

Customer Form

Enter customer id:

Submit Reset



### ***Customer Information***

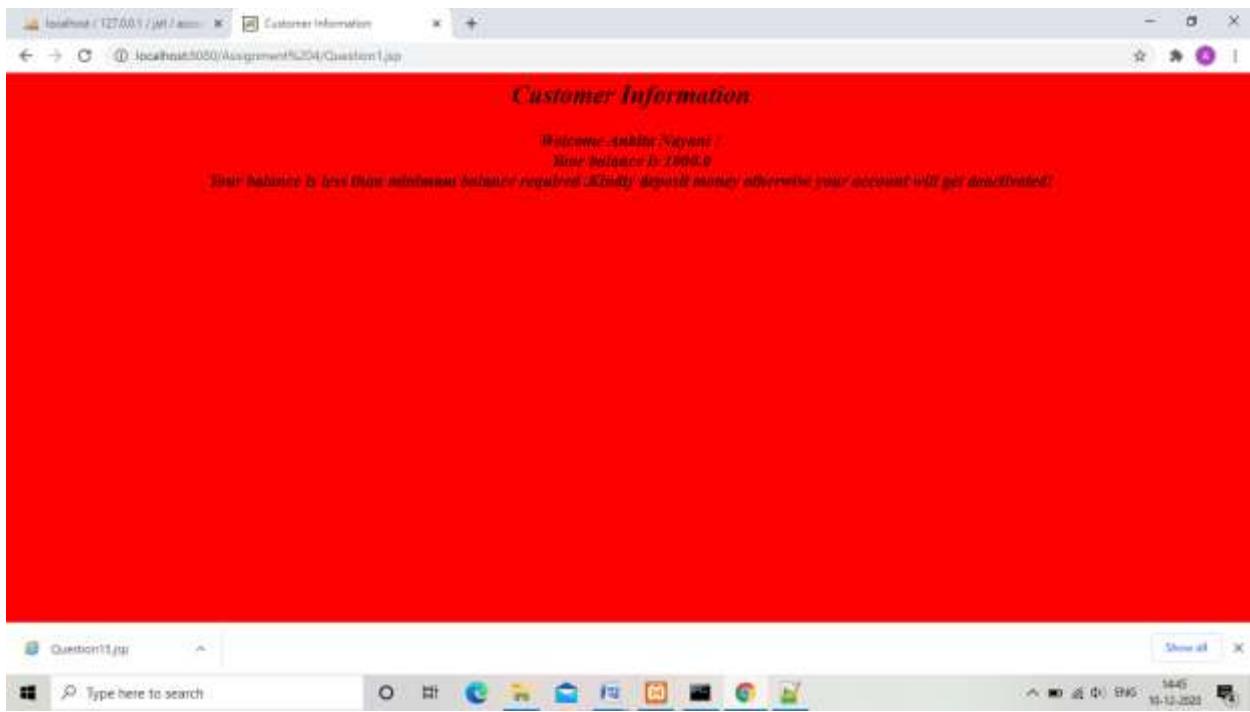
Welcome Antra koul!  
Your balance is 10000.85



### **Customer Form**

Enter customer id:





(b) Make an HTML page with a push button. When user clicks this button then shows him the content from a file. Update this after every 3 seconds.

“Question1bForm.html”

```
<html>
    <head>
        <title>user input</title>
    </head>
    <body>
        <center>
            <form method="post" action="Question1b.jsp">
                <h2><i>Click the push button to see file content!</i><br
            /><br />
                <input type="submit" value="push">
            </form>
        </center>
    </body>
</html>
```

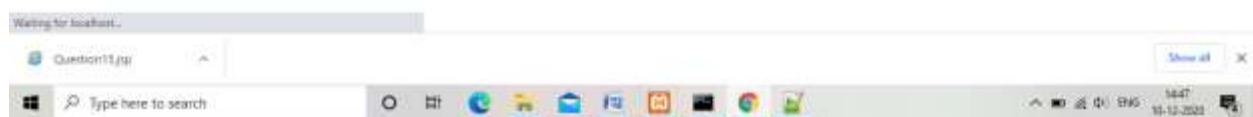
“Question1b.jsp”

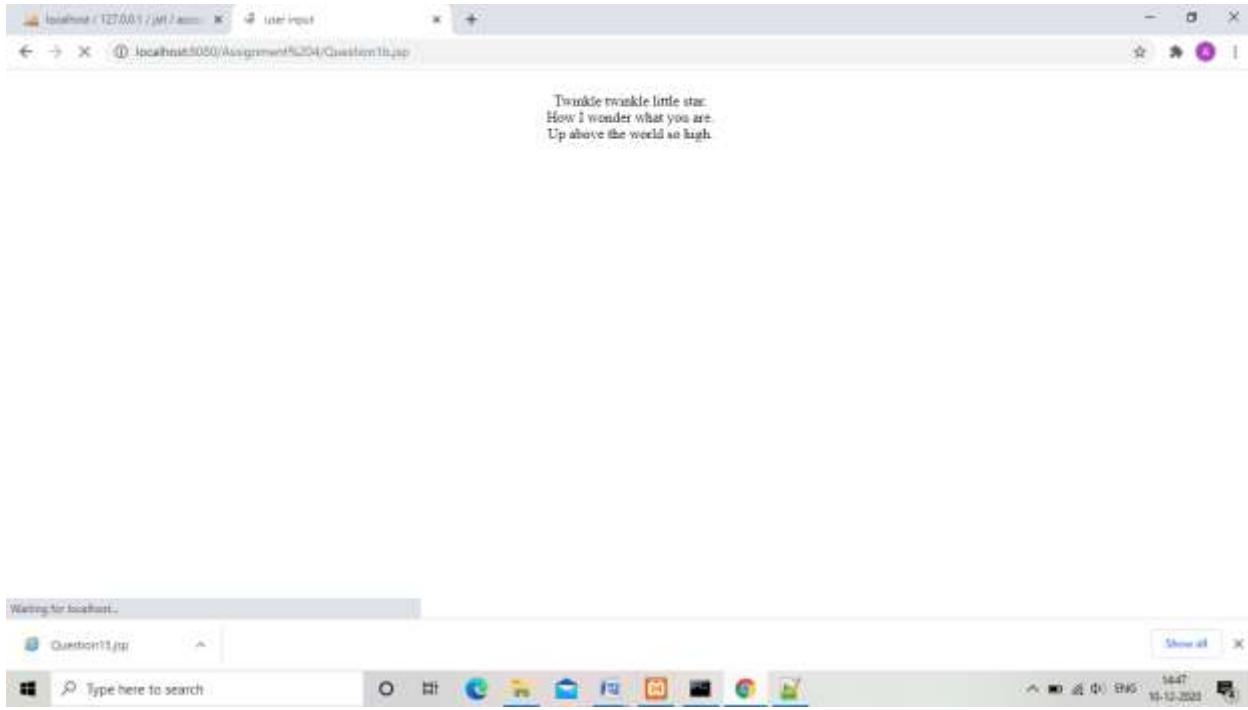
```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page import="java.io.*" %>
<html>
```

```
<head>
    <title>user input</title>
</head>
<body>
    <center>
        <%
            InputStream ins =
application.getResourceAsStream("/WEB-INF/File1.txt");
            BufferedReader br = new BufferedReader(new
InputStreamReader(ins));
            String line;
            while((line=br.readLine()) != null){
                Thread.sleep(3000);

            %>
            <c:set var="content" value="<%=" line%>" /><br />
            <c:out value="${content}" />
        <%
            out.flush();
        }
        %>
    </body>
</html>
```

“Output”





## “Ajax”

1. Make an HTML page with a pushbutton that invokes a alert box, which displays the either “Good Morning”, “Good Afternoon” or “Good Evening” based on server time using Ajax-Prototype framework.

### “Question2Form.html”

```
<html>
    <head>
        <title>Greeting message</title>
    </head>
    <script>
        var request;
        function showGreeting(){
            if(window.XMLHttpRequest){
                request = new XMLHttpRequest();
            }
            else if(window.ActiveXObject){
                request = new ActiveXObject("Microsoft.XMLHTTP");
            }
            try{
                request.onreadystatechange=getGreeting;
                request.open("GET","Question2.jsp",true);
            }
        }
        function getGreeting(){
            alert("Good Evening");
        }
    </script>
</html>
```

```

        request.send();
    }
    catch(e){
        alert("Unable to connect to the server!");
    }
}
function getGreeting(){
if(request.readyState==4){
    var val = request.responseText;
    alert(val);
}
}

</script>
<body>
<center>
<form>
<label>Press the button for greeting message:</label>
<input type="button" value="Push button"
onclick="showGreeting()">
</form>
</center>
</body>
</html>

```

“Question2.jsp”

```

<%@ page import = "java.util.Date" %>
<%
Date today = new Date();
int hours = today.getHours();
if(hours <= 12 && hours >= 0)
    out.print("Good morning!");
else if(hours <= 19 && hours >=13)
    out.print("Good afternoon!");
else
    out.print("Good evening!");
%>

```

“Output”



## “Assignment 5”

## 1. 1 Develop the following application:

A. Design an html page that accepts the father's name and age along with the son's name and age. The user should be redirected to "myError.jsp" if any of the age is invalid and also if father's age is less than son's age.

B. If the age is valid, the user should be redirected to "Welcome.jsp", where father's age at the time of son's birth is shown.

C. The valid details should be stored in the database using Statement or PreparedStatement

"Form.html"

```
<html>
    <head>
        <title>
            User Form
        </title>
    </head>
    <body>
        <center><h3><u>User Form</u></h3><br><br>
        <table>
            <form action="UserServlet" method="post">
                <tr>
                    <td>
                        <label>Enter father name:</label>
                    </td>
                    <td>
                        <input type="text" name="fathername"
/><br></br>
                    </td>
                </tr>
                <tr>
                    <td>
                        <label>Enter father age:</label>
                    </td>
                    <td>
                        <input type="text" name="fatherage"
/><br></br>
                    </td>
```

```

        </tr>
        <tr>
            <td>
                <label>Enter son name:</label>
            </td>
            <td>
                <input type="text" name="sonname"
/><br></br>
            </td>
        </tr>
        <tr>
            <td>
                <label>Enter son age:</label>
            </td>
            <td>
                <input type="text" name="sonage" /><br><br>
            </td>
        </tr>
        <tr>
            <td>
                <input type="submit" value="submit">
            </td>
            <td>
                <input type="reset" value="reset">
            </td>
        </tr>
    </form>
    </table>
</center>
</body>
</html>

```

“myerror.jsp”

```

<p>
    <h4><i>Sorry an error occurred!
</p>

```

“welcome.jsp”

```

<p>
    <h4><i>Father age at the time of son birth is :<%
    out.print(request.getAttribute("age"));
    %>

```

</p>

“web.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at


```

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

-->

```
<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>UserServlet</servlet-name>
    <servlet-class>fatherson.UserServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>UserServlet</servlet-name>
    <url-pattern>/UserServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

“FatherSondetails.java”

```
package fatherson;
public class FatherSonDetails{
    private String fathername,sonname;
    private int fatherage,sonage;
    public void setFathername(String fathername){
        this.fathername=fathername;
    }

    public String getFathername(){
        return fathername;
    }

    public void setSonname(String sonname){
        this.sonname=sonname;
    }

    public String getSonname(){
        return this.sonname;
    }

    public void setSonage(int sonage){
        this.sonage=sonage;
    }

    public int getSonage(){
        return this.sonage;
    }

    public void setFatherage(int fatherage){
        this.fatherage=fatherage;
    }

    public int getFatherage(){
        return this.fatherage;
    }
}
```

“userServlet.java”

```
package fatherson;

import java.sql.*;
import java.io.IOException;
```

```

import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class UserServlet extends HttpServlet{

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int fatherage=Integer.parseInt(request.getParameter("fatherage"));
        int sonage=Integer.parseInt(request.getParameter("sonage"));
        if( fatherage< 0 || sonage< 0 || fatherage < sonage)
        {
            RequestDispatcher request_dispatcher =
request.getRequestDispatcher("myError.jsp");
            request_dispatcher.forward(request,response);
        }
        else{
            try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/jstl","root","");
                PreparedStatement stmt =
connection.prepareStatement("Insert into FatherSon values(?,?,?,?,?)");
                stmt.setString(1,request.getParameter("fathername"));
                stmt.setInt(2,fatherage);
                stmt.setString(3,request.getParameter("sonname"));
                stmt.setInt(4,sonage);

                int i =stmt.executeUpdate();
                System.out.print("Record:"+i);
                connection.close();
            }
            catch(Exception e){
                e.printStackTrace();
            }
            int age=fatherage-sonage;

```

```

        request.setAttribute("age",age+"");
        RequestDispatcher request_dispatcher =
request.getRequestDispatcher("welcome.jsp");
        request_dispatcher.forward(request,response);
    }

}

protected void doGet(HttpServletRequest request,HttpServletResponse response)
throws ServletException, IOException {
    doPost(request, response);
}
}

```

"Output"

Fathername	Fatherage	Sonname	Sonage
Tom	65	Jerry	12
Harry	65	Ben	12
Harry	39	Ben	12
Ross	48	Ben	18
Garry	65	Jerry	23
Tom	65	Ben	12
Ross	58	Jerry	12
Ross	58	Jerry	12

localhost : 127.0.0.1 / JAI / father - X User Form

← → C ① localhost:8080/WebApplication1/Form.html

User Form

Enter father name: Tim

Enter father age: 50

Enter son name: Penny

Enter son age: 12

Question1.jsp

Type here to search

localhost : 127.0.0.1 / JAI / father - X User Form

← → C ① localhost:8080/WebApplication1/UserService

Father age at the time of son birth is :38

Question1.jsp

Type here to search

localhost : 127.0.0.1 / JAI / father - X User Form

← → C ① localhost:8080/WebApplication1/UserService

User Form

Enter father name:

Enter father age:

Enter son name:

Enter son age:



2. 3. Develop the following application:

- A. Design an MVC application where-in user enter his user-id in an html form "index.jsp".

- B. Based on user-id, the user's data should be fetched from the database.
- C. If the user-type is of "student", he should be redirected to "subjects.jsp" and if the user-type is "faculty", he should be redirected to "profile.jsp".
- D. Maintain a count for how many time the "index.jsp" has been accessed from the server reboot

"index.jsp"

```

<html>
  <head>
    <title>
      User Form
    </title>
  </head>
  <body>
    <%
      String count=(String)application.getAttribute("visitor_number");
      if(count==null)
      {
        application.setAttribute("visitor_number","1");
        count=(String)application.getAttribute("visitor_number");
      }
      else
      {
        int counter=Integer.parseInt(count);
        counter=counter+1;
        application.setAttribute("visitor_number",counter+"");
      }
    %>
  </body>
</html>
```

```

        count=(String)application.getAttribute("visitor_number");

    }

    out.print("Visitor number:"+count);

%>

<center><h3><u>User Form</u></h3><br><br>

<table>

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

<tr>

<td>

<label>Enter user id :</label>

</td>

<td>

<input type="text" name="userid" /><br><br>

</td>

</tr>

<tr>

<td>

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

</td>

<td>

<input type="reset" value="reset">

</td>

</tr>

</form>

</table>

```

```

        </center>

    </body>

</html>

        "profile.jsp"

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<html>

    <head>

        <title>

            Subjects

        </title>

    </head>

    <body>

        <h3><i>Teacher Profile

        <c:set var="id" value="${requestScope['id']}" />

        <sql:setDataSource var="db" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/jstl" user="root" password="">

        <sql:query dataSource="${db}" var="rs">

            Select * from profile ;

        </sql:query>

        <table width="50%" height="50%">

            <tr>

                <th>Qualification</th>

                <th>Salary</th>

            </tr>

        <c:forEach var="table" items="${rs.rows}">

```

```

<c:set var="userid" value="${table.ID}" />

<c:if test="${userid==id}">

<tr>

<td align="center"><c:out value="${table.Qualification}" /></td>

<td align="center"><c:out value="${table.Salary}" /></td>

</tr>

</c:if>

</c:forEach>

</table>

</body>

</html>

```

“subjects.jsp”

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<html>

<head>

<title>

    Subjects

</title>

</head>

<body>

    <h3><i>Subjects:</i></h3>

    <c:set var="id" value="${requestScope['id']}' />

    <sql:setDataSource var="db" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/jstl" user="root" password="" />

```

```

<sql:query dataSource="${db}" var="rs">

    Select * from subject ;

</sql:query>

<table width="50%" height="50%">

    <tr>

        <th>Subject</th>

    </tr>

    <c:forEach var="table" items="${rs.rows}">

        <c:set var="userid" value="${table.ID}" />

        <c:if test="${userid==id}">

            <tr>

                <td align="center"><c:out value="${table.subjects}" /></td>

            </tr>

        </c:if>

    </c:forEach>

    </table>

</body>

</html>

```

“IDServlet.java”

```

package userid;

import java.sql.*;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class IDServlet extends HttpServlet{

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int id=Integer.parseInt(request.getParameter("userid"));
        request.setAttribute("id",id+"");
        String type="";
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/jstl","root","");
            Statement stmt = connection.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from user");
            while(rs.next()){
                if(rs.getInt("ID") == id){
                    type=rs.getString("Type");
                    break;
                }
            }
        }
    }
}
```

```

        }

        if(type.equalsIgnoreCase("Student")){
            RequestDispatcher request_dispatcher =
request.getRequestDispatcher("subjects.jsp");
            request_dispatcher.forward(request,response);

        }

        else

        {

            RequestDispatcher request_dispatcher =
request.getRequestDispatcher("profile.jsp");
            request_dispatcher.forward(request,response);

        }

        connection.close();

    }

    catch(Exception e){

        e.printStackTrace();

    }

}

protected void doGet(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {

    doPost(request, response);

}

}

“web.xml”

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

```

contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

-->

```
<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>
```

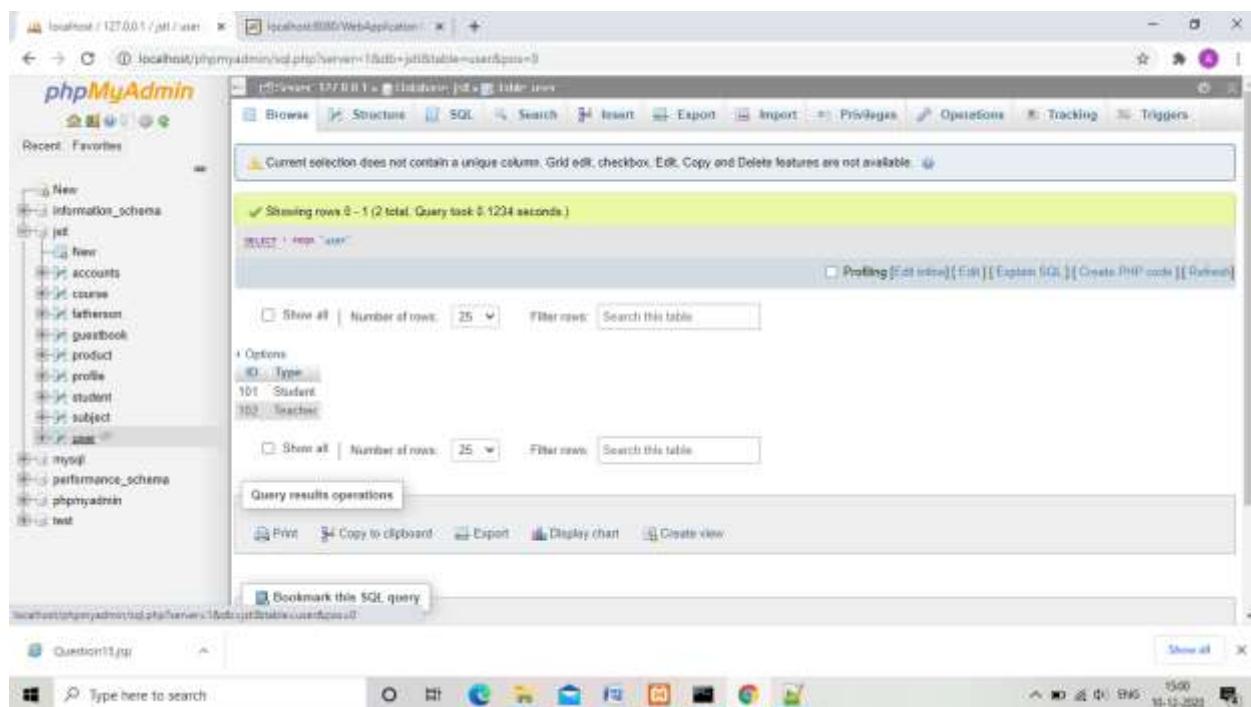
```
    <description>
```

```
      Welcome to Tomcat
```

```
    </description>
```

```
<servlet>  
    <servlet-name>IDServlet</servlet-name>  
  
    <servlet-class>userid.IDServlet</servlet-class>  
  
</servlet>  
  
<servlet-mapping>  
    <servlet-name>IDServlet</servlet-name>  
  
    <url-pattern>/IDServlet</url-pattern>  
  
</servlet-mapping>  
  
</web-app>
```

## “Output”



localhost / 127.0.0.1 / jdb / subj -> localhost:8080/WebApplication -> +

localhost/phpmyadmin/sql.php?server=1&db=jdb&table=subject&pos=0

**phpMyAdmin**

Recent: Favorites

New

information\_schema

jdb

  New

  accounts

  course

  teacher

  guestbook

  product

  profile

  student

  subject

  user

mysql

performance\_schema

phpmyadmin

test

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total. Query took 0.0007 seconds.)

SELECT \* FROM `subject`

Profiling [Edit trace] [Edit] [Export SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Options

ID: Subjects

Hindi

101 English

Math

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

localhost/phpmyadmin/sql.php?server=1&db=jdb&table=subject&pos=0

Question11.jsp

Type here to search

localhost / 127.0.0.1 / jdb / prof -> localhost:8080/WebApplication -> +

localhost/phpmyadmin/sql.php?server=1&db=jdb&table=profedges&pos=0

**phpMyAdmin**

Recent: Favorites

New

information\_schema

jdb

  New

  accounts

  course

  teacher

  guestbook

  product

  profile

  student

  subject

  user

mysql

performance\_schema

phpmyadmin

test

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total. Query took 0.0007 seconds.)

SELECT \* FROM `profile`

Profiling [Edit trace] [Edit] [Export SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Options

ID: Qualification

102 MBA 210000

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

Bookmark this SQL query

localhost/phpmyadmin/sql.php?server=1&db=jdb&table=profedges&pos=0

List every user access this bookmark

Question11.jsp

Type here to search





3.

4. Develop an application for a GuestBook as following:

- A. Design an HTML page that takes user's name, email and the comments of the user.  
 The page should have three buttons submit, clear form and view guestbook
- B. On the click of submit the user's details should be stored in the database and the user should be shown a page showing his entered details and a note of thanks.
- C. On clicking the clear form button the data entered on the form should be cleared. And on clicking on view Guestbook page the whole guest book should be shown to the user.

"index.jsp"

<html>

<head>

<title>

User Form

</title>

</head>

<body>

<center><h3><u>Guest  
 Form</u></h3><br><br>

<table>

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

<tr>

<td>

<label>Enter user name:</label>

</td>

<td>

<input  
 type="text" name="username" /><br></br>

</td>

</tr>

<tr>

<td>

<label>Enter email:</label>

</td>

<td>

<input

type="text" name="email" /><br></br>

</td>

</tr>

<tr>

<td>

<label>Enter comments:</label>

</td>

<td>

<textarea rows="5" cols="20"  
name="comments"></textarea><br></br>

</td>

</tr>

<tr>

<td>

<input

type="submit" value="submit">

</td>

```

<td>

<input
type="reset" value="reset">

</td>

</tr>

</form>

</table>

<form action="display.jsp">

<input type="submit"
value="View Guest book">

</form>

</center>

</body>

</html>

"ShowDetail.jsp"

<html>

<head>

<title>

User Form

</title>

</head>

<body>

<center><h3><u>Thanks for your feedback and entry!</u></h3><br><br>

Entered Details:<%= request.getAttribute("Detail")%>

</center>

</body>

```

```

</html>

        "display.jsp"

<%@ page import="java.io.* ,java.util.* ,java.sql.*"%>

<%@ page import="javax.servlet.http.* ,javax.servlet.*" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>

    <head>

        <title>

            Student Information

        </title>

    </head>

    <body><h3>GuestBook

        <sql:setDataSource var="db" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/jstl" user="root" password="" />

        <sql:query dataSource="${db}" var="rs">

            Select * from Guestbook

        </sql:query>

        <table >

            <tr>

                <th>Username</th>

                <th>Email</th>

                <th>Comments</th>

            </tr>

            <c:forEach var="table" items="${rs.rows}">

                <tr>

```

```

        <td align="center"><c:out value="${table.Username}" /></td>
        <td align="center"><c:out value="${table.Email}" /></td>
        <td align="center"><c:out value="${table.Comments}" /></td>
    </tr>
</c:forEach>
</table>

</body>
</html>

```

“GuestServlet.java”

```

package guest;

import java.sql.*;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class GuestServlet 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 connection =
DriverManager.getConnection("jdbc:mysql://localhost/jstl","root","");
    PreparedStatement stmt = connection.prepareStatement("Insert into
Guestbook values(?, ?, ?)");

    stmt.setString(1,request.getParameter("username"));

    stmt.setString(2,request.getParameter("email"));

    stmt.setString(3,request.getParameter("comments"));

    String
detail="Username:"+request.getParameter("username")+"\nEmail"+request.getParameter("email")+"\n
Comments"+request.getParameter("comments");

    request.setAttribute("Detail",detail);

    int i =stmt.executeUpdate();

    System.out.print("Record:"+i);

    connection.close();

}

catch(Exception e){

    e.printStackTrace();

}

RequestDispatcher request_dispatcher =
request.getRequestDispatcher("ShowDetail.jsp");

request_dispatcher.forward(request,response);

```

```
}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    doPost(request, response);
}

}

“web.xml”
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
```

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

-->

```
<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>GuestServlet</servlet-name>  
    <servlet-class>guest.GuestServlet</servlet-class>  
</servlet>  
<servlet-mapping>  
    <servlet-name>GuestServlet</servlet-name>  
    <url-pattern>/GuestServlet</url-pattern>  
</servlet-mapping>  
</web-app>
```

“Output”

localhost / 127.0.0.1 / MySQL / guest

Subjects

phpMyAdmin

Recent: Favorites

New

information\_schema

jeft

new

accounts

course

featheron

guestbook

product

profile

student

subject

user

mysql

performance\_schema

phpmyadmin

test

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 3 - 6 (6 total. Query took 0.023 seconds.)

SELECT \* FROM `guestbook`

Show all Number of rows: 25 Filter rows Search this table

+ Options

username	Email	Comments
Anita kud	anitakud1985@gmail.com	Very nice hospitality
Anita kud	anitakud1985@gmail.com	Very nice hospitality
Anilka	anilka@gmail.com	good service
Aishish	aishish@gmail.com	Good services
Aishish	aishish@gmail.com	Good services
Aishish	aishish@gmail.com	Good services

Show all Number of rows: 25 Filter rows Search this table

Query results operations

Console Copy to clipboard Export Display chart Create view

Show all

The screenshot shows the phpMyAdmin interface for a MySQL database named 'jeft'. The 'guestbook' table is selected. The table has three columns: 'username', 'Email', and 'Comments'. There are six rows of data. The first two rows have the same email address ('anitakud1985@gmail.com') and comments ('Very nice hospitality'). The other four rows have unique email addresses ('anilka@gmail.com', 'aishish@gmail.com', 'aishish@gmail.com', and 'aishish@gmail.com') and comments ('good service', 'Good services', 'Good services', and 'Good services' respectively). The interface includes a sidebar with various databases and tables, and a bottom navigation bar with options like 'Console', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

localhost / 127.0.0.1 / MySQL / guest

User Form

Guest Form

Enter user name: Kruti

Enter email: kruti@gmail.com

Very good service!

Enter comments:

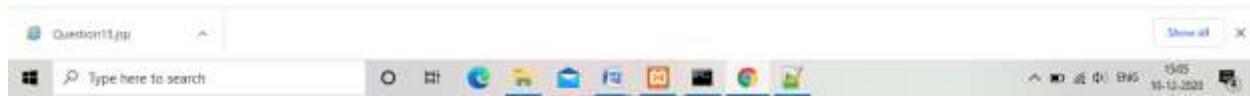
submit reset View Guestbook

The screenshot shows a web browser window titled 'User Form' with a sub-section titled 'Guest Form'. It contains three text input fields: 'Enter user name:' with the value 'Kruti', 'Enter email:' with the value 'kruti@gmail.com', and a text area for 'Enter comments:' containing the text 'Very good service!'. Below the inputs are three buttons: 'submit', 'reset', and 'View Guestbook'.





Entered Details: Username: KrutiEmail:kruti@gmail.comComments:Very good service



GuestBook		
Username	Email	Comments
Antra koul	antrakoul1183@gmail.com	Very nice hospitality
Antra koul	antrakoul1183@gmail.com	Very nice hospitality
Akita	akita@gmail.com	good service
Ashish	ashish@gmail.com	Good services
Ashish	ashish@gmail.com	Good services
Ashish	ashish@gmail.com	Good services
Kruti	kruti@gmail.com	Very good service



4.

6. Develop an application for a Inventory for Products as following:

A. Design an HTML page that takes the product details like:

Product-id  
Product-name  
Desctiption  
Price from the user

- B. On submitting the page a bean should be populated with the given values for Product in a servlet
- C. The bean created then should be inserted in the database.
- D. On the same page, all the available products in the database should be shown to the user.

“Form.html”

```
<html>
  <head>
    <title>Product Form</title>
  </head>
  <body>
    <center>
      <h2><u>Product form Form</u><br /><br />
      <form method="post" action="Question3">
        <label>Enter product id:</label>
        <input type="text" name="productid"><br /><br />
        <label>Enter product name:</label>
        <input type="text" name="productname"><br /><br />
        <label>Enter description:</label>
```

```

<input type="text" name="description"><br /><br />

<label>Enter price:</label>

<input type="text" name="price"><br /><br />

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

<input type="reset" value="reset">

</form>

</center>

</body>

</html>

```

“Product.jsp”

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<html>

<head>
    <title>Product details</title>
</head>
<body>

    <sql:setDataSource var="db" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/jstl" user="root" password="" />

    <sql:query var="rs" dataSource="${db}">

        Select * from Product;

    </sql:query>

    <center><h3><i>Product data</i></h3>
    <table>

        <tr>
            <th>Product ID</th>

```

```

<th>Product name</th>
<th>Product Description</th>
<th>Product price</th>
</tr>
<c:forEach var="table" items="${rs.rows}">
<tr>
<td align="center"><c:out value="${table.ID}" /></td>
<td align="center"><c:out value="${table.Name}" /></td>
<td align="center"><c:out value="${table.Description}" /></td>
<td align="center"><c:out value="${table.Price}" /></td>
</tr>
</c:forEach>
</table>
</center>
</body>
</html>
```

“ProductDetails.java”

```

package product;

public class ProductDetails{
    private String productid,productname,description,price;
    public void setProductid(String productid){
        this.productid=productid;
    }
    public String getProductid(){
```

```
    return productid;  
}  
  
  
public void setProductname(String productname){  
    this.productname=productname;  
}  
  
  
public String getProductname(){  
    return this.productname;  
}  
  
  
public void setDescription(String description){  
    this.description=description;  
}  
  
  
public String getDescription(){  
    return description;  
}  
  
  
public String getPrice(){  
    return this.price;  
}  
  
  
public void setPrice(String price){  
    this.price=price;
```

```
    }  
}  
  
"Question3.java"
```

```
package product;
```

```
import java.sql.*;  
  
import java.io.IOException;  
  
import java.io.PrintWriter;  
  
import javax.servlet.RequestDispatcher;  
  
import javax.servlet.ServletException;  
  
import javax.servlet.http.HttpServlet;  
  
import javax.servlet.http.HttpServletRequest;  
  
import javax.servlet.http.HttpServletResponse;
```

```
public class Question3 extends HttpServlet{
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
        response.setContentType("text/html");  
  
        PrintWriter out = response.getWriter();
```

```
        ProductDetails p1 = new ProductDetails();  
  
        p1.setProductid(request.getParameter("productid"));  
  
        p1.setProductname(request.getParameter("productname"));  
  
        p1.setDescription(request.getParameter("description"));
```

```

p1.setPrice(request.getParameter("price"));

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection connection =
    DriverManager.getConnection("jdbc:mysql://localhost/jstl","root","");
    PreparedStatement stmt = connection.prepareStatement("Insert into Product
values(?, ?, ?, ?)");
    stmt.setString(1,request.getParameter("productid"));
    stmt.setString(2,request.getParameter("productname"));
    stmt.setString(3,request.getParameter("description"));
    stmt.setString(4,request.getParameter("price"));

    int i =stmt.executeUpdate();
    System.out.print("Record:"+i);
    connection.close();
}
catch(Exception e){
    e.printStackTrace();
}

RequestDispatcher request_dispatcher = request.getRequestDispatcher("Products.jsp");
request_dispatcher.forward(request,response);
}

protected void doGet(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
    doPost(request, response);
}

```

```
}

}

"web.xml"
```

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

<!--

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at


```

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

-->

```
<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>Question3</servlet-name>
    <servlet-class>product.Question3</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Question3</servlet-name>
    <url-pattern>/Question3</url-pattern>
</servlet-mapping>

</web-app>
```

“Output”

localhost / 127.0.0.1 / MySQL / phpMyAdmin / Student Information

localhost/phpmyadmin/13.php?server=1&db=jotitstable&tbl=product

**phpMyAdmin**

Recent: Favorites

New

information\_schema

jotit

New

accounts

course

latherson

guestbook

**product**

profile

student

subject

user

mysql

performance\_schema

phpmyadmin

test

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total. Query took 0.0007 seconds.)

SELECT \* FROM `product`

Profiling (Edit) (Exit) (Export SQL) (Create PHP code) (Downtime)

Show all Number of rows: 25 Filter rows: Search this table

ID	Name	Description	Price
101	pen	a cold pen	123
102	Notepad	descriptive	123
103	Pencil	Apsara Dark	34

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

localhost/phpmyadmin/13.php?server=1&db=jotitstable&tbl=product&pos=0

Question11.jsp

Show all

localhost / 127.0.0.1 / MySQL / phpMyAdmin / Product Form

localhost:8080/WebApplication01/Form.html

### Product form Form

Enter product id:

Enter product name:

Enter description:

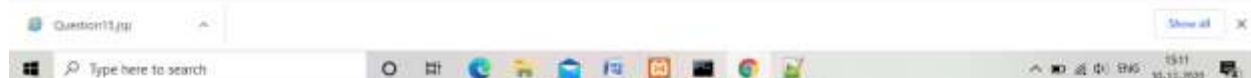
Enter price:

Add product reset

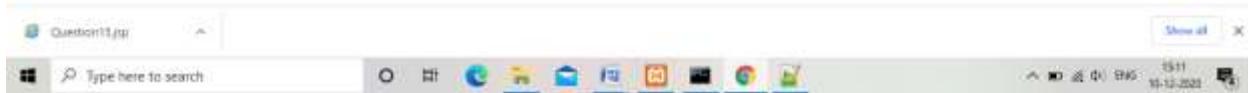
localhost / 127.0.0.1 / MySQL / phpMyAdmin / Question11.jsp

Type here to search

15:10 15-12-2021



Product data				
Product ID	Product name	Product Description	Product price	
101	pen	a cello pen	123	
102	Notebook	classmate	123	
103	Pencil	Apsara Dark	34	
104	colors	Apsara	30	



5. Develop an application as following:

- A. Design an HTML form that takes the input as user-id from the user.
- B. Based on the id, fetch the data from the database. Check his type and accordingly populate the bean of MCASStudent or MBASStudent.
- C. If the user is an MCASStudent, he should be given a page that displays the subject names for MCA and if the user is an MBASStudent, the subject should be shown

accordingly from the database.

“index.jsp”

<html>

<head>

<title>

User Form

```
</title>

</head>

<body>

<center><h3><u>User Form</u></h3><br><br>

<table>

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

<tr>

<td>

<label>Enter user id :</label>

</td>

<td>

<input type="text" name="userid" /><br><br>

</td>

</tr>

<tr>

<td>

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

</td>

<td>

<input type="reset" value="reset">

</td>

</tr>

</form>

</table>
```

```

        </center>

    </body>

</html>

        “mba.jsp”

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page import="userid.MBA"%>

<html>

    <head>

        <title>

            MBA Subjects

        </title>

    </head>

    <body>

        <%
            MBA m=(MBA)request.getAttribute("MBA");
        %>

        <center><i>

            Subject1:<%= m.getSubject1()%><br /><br />
            Subject2:<%= m.getSubject2()%><br /><br />
            Subject3:<%= m.getSubject3()%></i></center>

    </body>

</html>

        “mca.jsp”

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

```

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page import="userid.MCA"%>
<html>

    <head>
        <title>
            MCA Subjects
        </title>
    </head>
    <body>
        <%
            MCA m=(MCA)request.getAttribute("MCA");
        %>
        <center><i>
            Subject1:<%= m.getSubject1()%><br /><br />
            Subject2:<%= m.getSubject2()%><br /><br />
            Subject3:<%= m.getSubject3()%></i></center>
    </body>
</html>

```

“web.xml”

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with

```

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

```
<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>StudentServlet</servlet-name>
<servlet-class>userid.StudentServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>StudentServlet</servlet-name>
<url-pattern>/StudentServlet</url-pattern>
</servlet-mapping>
</web-app>
```

“MBA.java”

```
package userid;
public class MBA{
    String subject1,subject2,subject3;

    public void setSubject1(String subject1){
        this.subject1=subject1;
    }
    public String getSubject1(){
        return this.subject1;
    }
    public void setSubject2(String subject2){
        this.subject2=subject2;
    }
    public String getSubject2(){
        return this.subject2;
    }
}
```

```
public void setSubject3(String subject3){  
    this.subject3=subject3;  
}  
  
public String getSubject3(){  
    return this.subject3;  
}  
}  
  
“MCA.java”  
  
package userid;  
  
public class MCA{  
    String subject1,subject2,subject3;  
  
    public void setSubject1(String subject1){  
        this.subject1=subject1;  
    }  
  
    public String getSubject1(){  
        return this.subject1;  
    }  
  
    public void setSubject2(String subject2){  
        this.subject2=subject2;  
    }  
  
    public String getSubject2(){  
        return this.subject2;  
    }  
  
    public void setSubject3(String subject3){  
        this.subject3=subject3;  
    }
```

```
    }

    public String getSubject3(){

        return this.subject3;

    }

}

“StudentServlet.java”
```

```
package userid;
```

```
import java.sql.*;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
```

```
public class StudentServlet extends HttpServlet{
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        int id=Integer.parseInt(request.getParameter("userid"));

        String type="";
```

```

try{

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

    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/jstl","root","");
}

Statement stmt = connection.createStatement();

ResultSet rs=stmt.executeQuery("Select * from course");

while(rs.next()){

    if(rs.getInt("ID") == id){

        type=rs.getString("CourseType");

        break;

    }

}

if(type.equalsIgnoreCase("MBA")){

    MBA m = new MBA();

    m.setSubject1("Finance");

    m.setSubject2("Operations");

    m.setSubject3("Marketing");

    request.setAttribute("MBA",m);

}

RequestDispatcher request_dispatcher =
request.getRequestDispatcher("mba.jsp");

request_dispatcher.forward(request,response);

}

else

{

    MCA m = new MCA();

    m.setSubject1("C");
}

```

```
        m.setSubject2("Java");

        m.setSubject3("Software Engineering");

        request.setAttribute("MCA",m);

        RequestDispatcher request_dispatcher =
request.getRequestDispatcher("mca.jsp");

        request_dispatcher.forward(request,response);

    }

    connection.close();

}

catch(Exception e){

    e.printStackTrace();

}

}

protected void doGet(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {

    doPost(request, response);

}

}

“Output”
```

localhost / 127.0.0.1 / MySQL / user -> Product details

localhost/phpmyadmin/123.php?server=1&db=jotitables&table=courseflipm

**phpMyAdmin**

Recent: Favorites

New

information\_schema

jotitables

accounts

courses

letherson

guesbook

product

profile

student

subject

user

mysql

performance\_schema

phpmyadmin

test

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 12 (total: 12 total. Query took 0.1116 seconds.)

SELECT \* FROM `course`

Profiling Edit inline Explain SQL Create PHP code Refresh

Show all Number of rows: 25 Filter rows Search this table Sort by key None

Options ID CourseType

101 MBA

102 MCA

Edit Copy Delete

Check all With selected Edit Copy Delete Export

Show all Number of rows: 25 Filter rows Search this table Sort by key None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

localhost/phpmyadmin/123.php?server=1&db=jotitables&table=courseflipm

Question11.jsp

Show all

localhost / 127.0.0.1 / MySQL / user -> UserForm

localhost:8080/WebApplication8/index.jsp

### User Form

Enter user id:

Question11.jsp

Show all



