IWP LAB-FAT

**How to Select Individual Records From MySQL Table Tutorial**

Learn how to select particular records from a MySQL table

As well as showing the whole database table, PHP can be used to select individual records, or records which match certain criteria. To do this you must use a variation of the **SELECT** query. To display the whole table you can use:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM tablename |

If you just want to select records which have *value=1* in the *field1-name* row you would use the following query:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM tablename WHERE field1-name='1' |

In the same way you could select records based on any field in the database. You can also search in more fields by adding more:

|  |  |
| --- | --- |
| 1 | field='value' |

sections into the query.

For example, the following query will select all records which have *value=1* in the *field1-name* row and *value=2* in the *field2-name* row:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM tablename WHERE field1-name='1' AND field2-name='2' |

For further reference you can visit the official websites of [PHP](http://php.net/) and [MySQL](http://mysql.com/).

**How to Query Your MySQL Database Tutorial**

Learn how to execute MySQL queries via PHP

If you want to use PHP to query your MySQL database you can do that by either entering the MySQL query command in the PHP script or define the command as a variable and use the variable when needed.

This tutorial explains how to perform the query by using a variable. In that case, the command will look similar to:

|  |  |
| --- | --- |
| 1 | mysqli\_query($query); |

The command can be repeated again in the source code. All you need to do is to change the *$query*variable.

For example, here is the complete code that could be used to create a MySQL table in PHP:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <?php  $username = "your\_username";  $password = "your\_password";  $database = "your\_database";  $mysqli = new mysqli("localhost", $username, $password, $database);  $query="CREATE TABLE tablename(id int(6) NOT NULL auto\_increment,first varchar(15) NOT NULL,last varchar(15) NOT NULL,field1-name varchar(20) NOT NULL,field2-name varchar(20)NOT NULL,field3-name varchar(20) NOT NULL,field4-name varchar(30) NOT NULL, field5-name varchar(30)NOT NULL,PRIMARY KEY (id),UNIQUE id (id),KEY id\_2 (id))";  $mysqli->query("$query");  $mysqli->close();  ?> |

Replace your *your\_username*, *your\_password* and *your\_database* with their actual values in the first three lines of the script.

You can replace the value of the *$query* variable with any MySQL query you want and you can use the above format to execute it.

**How to Display MySQL Table Data Tutorial**

Learn how to populate and display MySQL table data with PHP

Very often you will need to use a MySQL table to store data inside it and then output that data by using a PHP script. To display the table data it is best to use HTML, which upon filling in some data on the page invokes a PHP script which will update the MySQL table.

To populate a new database table with data you will first need an HTML page which will collect that data from the user. The following HTML code that and passes the information to a PHP script:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <form action="insert.php" method="post">  Value1: <input type="text" name="field1-name" />  Value2: <input type="text" name="field2-name" />  Value3: <input type="text" name="field3-name" />  Value4: <input type="text" name="field4-name" />  Value5: <input type="text" name="field5-name" />  <input type="Submit" /></form> |

The above HTML code will show the user **5 text fields**, in which the user can input data and a **Submit** button. Upon clicking the **Submit** button the data submitted by the user will be passed to a script named **insert.php**.

That script can have a syntax similar to the following:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?php  $username = "your\_username";  $password = "your\_password";  $database = "your\_database";  $field1-name=$\_POST['Value1'];  $field2-name=$\_POST['Value2'];  $field3-name=$\_POST['Value3'];  $field4-name=$\_POST['Value4'];  $field5-name=$\_POST['Value5'];  $mysqli = new mysqli("localhost", $username, $password, $database);  @mysql\_select\_db($database) or die( "Unable to select database");  $query = "INSERT INTO tablename VALUES('','$field1-name','$field2-name','$field3-name','$field4-name','$field5-name')";  $mysqli->query($query);  $mysqli->close();  ?> |

After the user submits the information, the **insert.php** script will save it in the database table. Then you may want to output that information, so that the user can see it on the page. The first command you will need to use is the *SELECT FROM* MySQL statement that has the following syntax:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM tablename; |

This is a basic MySQL query which will tell the script to select all the records from the *tablename* table. After the query is executed, usually you would want the result from it stored inside a variable. This can be done with the following PHP code:

|  |  |
| --- | --- |
| 1  2 | $query2="SELECT \* FROM tablename";  $result=$mysqli->query($query2); |

The whole content of the table is now included in a PHP array with the name *$result*. Before you can output this data you should change each piece into a separate variable. There are two stages.

The first one is counting the rows. Before you can go through the data in your result variable, you should know the number of the database rows. You could, of course, just type this into your code but it is not a very good solution as the script code will have to be changed every time a new row is added. Instead you can use the command:

|  |  |
| --- | --- |
| 1 | $num=$mysqli->mysqli\_num\_rows($result); |

The *$num* value will be the number of rows stored in *$result*. This will be used in a loop to get all the data and display it on the screen.

The second stage is to set up the loop. It will take each row of the result and print the data stored there. In the code below, *$i*is the number of times the loop runs. This way all the records are displayed.

|  |  |
| --- | --- |
| 1  2  3  4  5 | $i=0;  while ($i < $num) {  CODE  $i++;  } |

This is a basic PHP loop and will execute *CODE* the correct number of times. Each time *$i* will be incremented by one. This is useful, as *$i* will tell the script which line of the results should be read. As the first line in MySQL output is 0, this will work correctly.

The final part of the output script is to assign each piece of data to its own variable:

|  |  |
| --- | --- |
| 1  2  3  4  5 | $field1-name=mysql\_result($result,$i,"field1-name");  $field2-name=mysql\_result($result,$i,"field2-name");  $field3-name=mysql\_result($result,$i,"field3-name");  $field4-name=mysql\_result($result,$i,"field4-name");  $field5-name=mysql\_result($result,$i,"field5-name"); |

You do not need to get the ID field because there is no use for it in the output page. You can now write a full script to output the data. In this script the data is not formatted when it is printed:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | <?php  $username="username";  $password="password";  $database="your\_database";  $mysqli = new mysqli("localhost", $username, $password, $database);  @mysql\_select\_db($database) or die( "Unable to select database");  $query2="SELECT \* FROM tablename";  $result=$mysqli->query($query2);  $num=$mysqli->mysqli\_num\_rows($result);  $mysqli->close();  echo "<b>  <center>Database Output</center>  </b>  <br>  <br>";  $i=0;  while ($i < $num) {  $field1-name=mysql\_result($result,$i,"field1-name");  $field2-name=mysql\_result($result,$i,"field2-name");  $field3-name=mysql\_result($result,$i,"field3-name");  $field4-name=mysql\_result($result,$i,"field4-name");  $field5-name=mysql\_result($result,$i,"field5-name");  echo "<b>  $field1-name $field2-name2</b>  <br>  $field3-name<br>  $field4-name<br>  $field5-name<hr>  <br>";  $i++;  }  ?> |

This outputs a list of all the values stored in the database. This will give you a very basic output which is not useful for a live website. Instead, it would be better if you could format it into a table and display the information in it. To apply formatting you need to use HTML to print the result by including the variables in the correct spaces. The easiest way to do this is by closing the PHP tag and entering HTML normally. When you reach a variable position, include it as follows:

|  |  |
| --- | --- |
| 1 | <? echo $variablename; ?> |

in the correct position in your code.

You can also use the PHP loop to repeat the appropriate code and include it as part of a larger table. The final output is:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | <html>  <body>  <?php  $username="username";  $password="password";  $database="your\_database";  $mysqli = new mysqli("localhost", $username, $password, $database); @mysql\_select\_db($database) or die( "Unable to select database");  $query2="SELECT \* FROM tablename";  $result=$mysqli->query($query2);  $num=$mysqli->mysqli\_num\_rows($result);?>  <table border="0" cellspacing="2" cellpadding="2">  <tr>  <td>  <font face="Arial, Helvetica, sans-serif">Value1</font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif">Value2</font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif">Value3</font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif">Value4</font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif">Value5</font>  </td>  </tr>  <?php  $i=0;  while ($i < $num) {  $f1=mysql\_result($result,$i,"field1");  $f2=mysql\_result($result,$i,"field2");  $f3=mysql\_result($result,$i,"field3");  $f4=mysql\_result($result,$i,"field4");  $f5=mysql\_result($result,$i,"field5");?>  <tr>  <td>  <font face="Arial, Helvetica, sans-serif"><?php echo $f1; ?></font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif"><?php echo $f2; ?></font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif"><?php echo $f3; ?></font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif"><?php echo $f4; ?></font>  </td>  <td>  <font face="Arial, Helvetica, sans-serif"><?php echo $f5; ?></font>  </td>  </tr>  <?php  $i++;  }?>  </body>  </html> |

This code will print out table content and add an extra row for each record in the database, formatting the data as it is printed.

FOrm validation:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Can Validate Input</h2>

<p>Please input a number between 1 and 10:</p>

<input id="numb">

<button type="button" onclick="myFunction()">Submit</button>

<p id="demo"></p>

<script>

function myFunction() {

var x, text;

// Get the value of the input field with id="numb"

x = document.getElementById("numb").value;

// If x is Not a Number or less than one or greater than 10

if (isNaN(x) || x < 1 || x > 10) {

text = "Input not valid";

} else {

text = "Input OK";

}

document.getElementById("demo").innerHTML = text;

}

</script>

</body>

</html>

## JavaScript Form Validation

HTML form validation can be done by JavaScript.

If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

### JavaScript Example

function validateForm() {  
    var x = document.forms["myForm"]["fname"].value;  
    if (x == "") {  
        alert("Name must be filled out");  
        return false;  
    }  
}

The function can be called when the form is submitted:

### HTML Form Example

<form name="myForm" action="/action\_page.php" **onsubmit="return validateForm()"** method="post">  
Name: <input type="text" name="fname">  
<input type="submit" value="Submit">  
</form>

**COMBINED CODE [ALL OF THE ABOVE SECTIONS CLUBBED)**

|  |
| --- |
| <html>  <head>  <script>  function GEEKFORGEEKS()  {      var name = document.forms["RegForm"]["Name"];      var email = document.forms["RegForm"]["EMail"];      var phone = document.forms["RegForm"]["Telephone"];      var what =  document.forms["RegForm"]["Subject"];      var password = document.forms["RegForm"]["Password"];      var address = document.forms["RegForm"]["Address"];        if (name.value == "")      {          window.alert("Please enter your name.");          name.focus();          return false;      }        if (address.value == "")      {          window.alert("Please enter your address.");          name.focus();          return false;      }        if (email.value == "")      {          window.alert("Please enter a valid e-mail address.");          email.focus();          return false;      }        if (email.value.indexOf("@", 0) < 0)      {          window.alert("Please enter a valid e-mail address.");          email.focus();          return false;      }        if (email.value.indexOf(".", 0) < 0)      {          window.alert("Please enter a valid e-mail address.");          email.focus();          return false;      }        if (phone.value == "")      {          window.alert("Please enter your telephone number.");          phone.focus();          return false;      }        if (password.value == "")      {          window.alert("Please enter your password");          password.focus();          return flase;      }        if (what.selectedIndex < 1)      {          alert("Please enter your course.");          what.focus();          return false;      }        return true;  }</script>    <style>  GEEKFORGEEKS {      margin-left: 70px;      font-weight: bold ;      float: left;      clear: left;      width: 100px;      text-align: left;      margin-right: 10px;      font-family:sans-serif,bold, Arial, Helvetica;      font-size:14px;  }    div {      box-sizing: border-box;      width: 100%;      border: 100px solid black;      float: left;      align-content: center;      align-items: center;  }    form {      margin: 0 auto;      width: 600px;  }</style></head>    <body>  <h1 style="text-align: center"> REGISTRATION FORM </h1>  <form name="RegForm" action="/submit.php" onsubmit="return GEEKFORGEEKS()" method="post">        <p>Name: <input type="text" size=65 name="Name"> </p><br>      <p> Address: <input type="text" size=65 name="Address">  </p><br>      <p>E-mail Address: <input type="text" size=65 name="EMail">  </p><br>       <p>Password: <input type="text" size=65 name="Password"> </p><br>      <p>Telephone: <input type="text" size=65 name="Telephone"> </p><br>        <p>SELECT YOUR COURSE          <select type="text" value="" name="Subject">              <option>BTECH</option>              <option>BBA</option>              <option>BCA</option>              <option>B.COM</option>              <option>GEEKFORGEEKS</option>          </select></p><br><br>      <p>Comments:  <textarea cols="55" name="Comment">  </textarea></p>      <p><input type="submit" value="send" name="Submit">          <input type="reset" value="Reset" name="Reset">      </p>  </form>  </body>  </html> |

NAMES

**ICQ UIN**

([1-9])+(?:-?\d){4,}

Top of Form



Bottom of Form

*by*[*Aurelian Hermand*](http://devone.de/)*&*[*Flobse*](irc://irc.quakenet.org/#regex)

**Alpha-Numeric**

[a-zA-Z0-9]+

Top of Form



Bottom of Form

*by*[*Aurelian Hermand*](http://devone.de/)

**Username with 2-20 chars**

^[a-zA-Z][a-zA-Z0-9-\_\.]{1,20}$

Format: string+string|number

Top of Form



Bottom of Form

*by*[*Aurelian Hermand*](http://devone.de/)

**Twitter Username**

^[A-Za-z0-9\_]{1,15}$

New Twitter usernames have a character maximum of 15

Top of Form



Bottom of Form

*by*[*Kevin Jones*](http://twitter.com/_kevinjones)

**Twitter Username**

^[A-Za-z0-9\_]{1,32}$

Backwards compatible Twitter username

Top of Form



Bottom of Form

*by*[*Kevin Jones*](http://twitter.com/_kevinjones)

**Facebook Username**

^[a-z\d\.]{5,}$

Top of Form



Bottom of Form

*by*[*Kevin Jones*](http://twitter.com/_kevinjones)

PASSWORDS

**Password (UpperCase, LowerCase and Number)**

^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?!.\*\s).\*$

Top of Form



Bottom of Form

*by*[*imar.spaanjaars.com*](http://imar.spaanjaars.com/297/regular-expression-for-a-strong-password)

**Password (UpperCase, LowerCase, Number/SpecialChar and min 8 Chars)**

(?=^.{8,}$)((?=.\*\d)|(?=.\*\W+))(?![.\n])(?=.\*[A-Z])(?=.\*[a-z]).\*$

Top of Form



Bottom of Form

*by*[*imar.spaanjaars.com*](http://imar.spaanjaars.com/297/regular-expression-for-a-strong-password)

MISCS

**IPv4 Address**

((^|\.)((25[0-5])|(2[0-4]\d)|(1\d\d)|([1-9]?\d))){4}$

Top of Form



Bottom of Form

*by Rasmus Fløe*

**IPv6 Address**

((^|:)([0-9a-fA-F]{0,4})){1,8}$

Top of Form



Bottom of Form

*by Rasmus Fløe & dipser*

**Domains like "abc.de"**

^([a-zA-Z0-9]([a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])?\.)+[a-zA-Z]{2,6}$

Top of Form



Bottom of Form

*by Unknown*

**Numbers with or without decimals**

[-+]?[0-9]\*[.,]?[0-9]+

Format: 9 or 9.9 or 9,9

Top of Form



Bottom of Form

*by Frédéric Hewitt*

**UUID**

^[0-9A-Fa-f]{8}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{12}$

Top of Form



Bottom of Form

*by blainsmith*

**Latitude or Longitude**

-?\d{1,3}\.\d+

Top of Form



Bottom of Form

*by*[*the-art-of-web.com*](http://www.the-art-of-web.com/html/html5-form-validation/)

**Price (Format: 1.00)**

\d+(\.\d{2})?

Top of Form



Bottom of Form

*by*[*the-art-of-web.com*](http://www.the-art-of-web.com/html/html5-form-validation/)

**Price (Format: 1,00)**

\d+(,\d{2})?

Top of Form



Bottom of Form

*by*[*the-art-of-web.com*](http://www.the-art-of-web.com/html/html5-form-validation/)

**ISBN**

(?:(?=.{17}$)97[89][ -](?:[0-9]+[ -]){2}[0-9]+[ -][0-9]|97[89][0-9]{10}|(?=.{13}$)(?:[0-9]+[ -]){2}[0-9]+[ -][0-9Xx]|[0-9]{9}[0-9Xx])

Combining of the following:  
\* Simple one without dashes, ISBN 13: 97[89][0-9]{10}  
\* Simple one w/o dashes, ISBN 10: [0-9]{9}[0-9Xx]  
  
\* Complex one with dashes, ISBN 13: (?=.{17}$)97[89]-(?:[0-9]+-){2}[0-9]+-[0-9]  
\* Complex one with dashes, ISBN 10: (?=.{13}$)(?:[0-9]+-){2}[0-9]+-[0-9Xx]

Top of Form



Bottom of Form

*by*[*Michael Ash*](http://regexlib.com/)*&*[*Boldewyn*](http://www.manuel-strehl.de/)

**Md5 Hash**

[0-9a-fA-F]{32}

Top of Form



Bottom of Form

*by*[*Ákos*](http://twitter.com/Yzahkin)

#### Username Patterns

Only letters (either case), numbers, and the underscore; no more than 15 characters. [A-Za-z0-9\_]{1,15}

Only lowercase letters and numbers; at least 5 characters, but no limit. [a-zd.]{5,}

Only letters (either case), numbers, hyphens, underscores, and periods. (Not the slash character, that is being used to escape the period.) The username must start with a letter and must be between 1 and 20 characters long (inclusive). [a-zA-Z][a-zA-Z0-9-\_.]{1,20}

#### Payment Info Patterns

USD Price Format (1.00) d+(.d{2})?

Credit Card Format - Digits only, between 13 and 16 digits long. [0-9]{13,16}

**EXPERIMENT 1**

**Invalid Input**

****

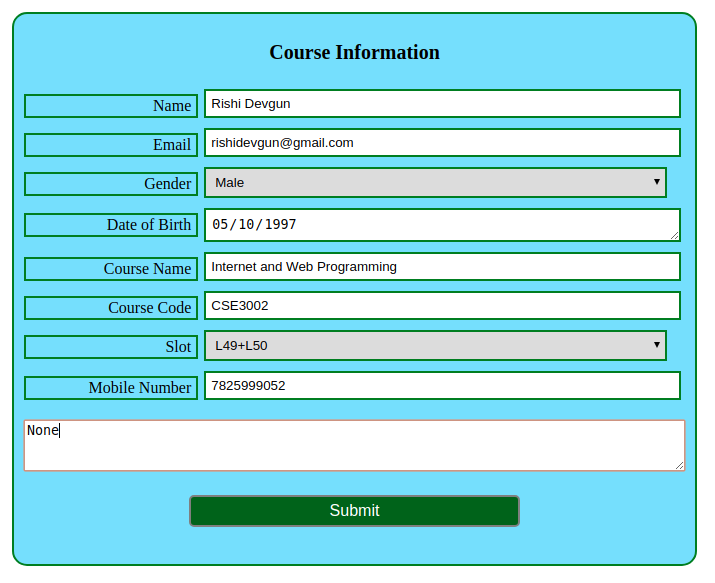
**Valid Input**

****

**Code:**

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <style> |
|  | body{ |
|  | background-color: gray; |
|  | text-align: center |
|  | } |
|  |  |
|  | h2{ |
|  | background-color: Tomato; |
|  | color : white; |
|  | padding-top = 0px; |
|  | font-size: 30px; |
|  | align: center; |
|  | border-top: 0px solid; |
|  | border-radius: 25px 25px 0px 0px; |
|  | border-bottom: thick solid Tomato; |
|  |  |
|  | } |
|  | .box{ |
|  | margin:100px 350px 0px; |
|  | padding-bottom:20px; |
|  | border: 0px solid; |
|  | border-radius: 25px; |
|  | background-color: #ffffff; |
|  | padding-top: 0px; |
|  | } |
|  |  |
|  | .textFields{ |
|  | width:450px; |
|  | padding:0px 0px 0px 50px ; |
|  | margin:15px 5px; |
|  | border: 1px inset; |
|  | font-family: "Times New Roman", Times, serif; |
|  | font-size:17px; |
|  | background-image: url(astrix.png); |
|  | background-size: 20px; |
|  | background-position:right center; |
|  | background-repeat: no-repeat; |
|  | } |
|  | input[type=text]:valid, input[type=email]:valid, input[type=password]:valid, input[type=url]:valid{ |
|  | background-color: #BFFF00; |
|  | background-image: url(valid.png); |
|  | background-size: 20px; |
|  | background-repeat: no-repeat; |
|  |  |
|  | } |
|  | input[type=text]:invalid:not(:placeholder-shown), input[type=email]:invalid:not(:placeholder-shown), input[type=password]:invalid:not(:placeholder-shown), input[type=url]:invalid:not(:placeholder-shown){ |
|  | border-color: red; |
|  | background-image: url(invalid.png); |
|  | background-size: 20px; |
|  | background-position:left center; |
|  | background-repeat: no-repeat; |
|  |  |
|  | } |
|  |  |
|  |  |
|  | input[type=button], input[type=submit], input[type=reset] { |
|  | background-color: Tomato; |
|  | border: none; |
|  | color: white; |
|  | padding: 5px 50px; |
|  | text-decoration: none; |
|  | margin: 4px 2px; |
|  | cursor: pointer; |
|  | font-family: "Times New Roman", Times, serif; |
|  | font-size:17px; |
|  | } |
|  |  |
|  | </style> |
|  | <html> |
|  | <body> |
|  | <div class = box> |
|  | <h2>HTML and CSS form Validation</h2> |
|  | <form> |
|  | <input class = textFields type="text" pattern = "[A-Za-z]{8,}" name="username" placeholder = "Username" required> |
|  | <br> |
|  | <input class = textFields type="password" pattern = "(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}" name="password" placeholder = "Password" required> |
|  | <br> |
|  | <input class = textFields type="email" pattern = "[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$" name="emailid" placeholder = "Email ID" required> |
|  | <br> |
|  | <input class = textFields type="text" pattern = "[0-9]{10}" name="MobileNumber" placeholder = "Mobile Number" required> |
|  | <br> |
|  | <input class = textFields type="url" pattern = "https?://.+" name="WebAddress" placeholder = "Web Address" required> |
|  | <br> |
|  | <input type="submit" value="Sign In"> |
|  | </form> |
|  | </div> |
|  | </body> |
|  | </html> |
|  |  |

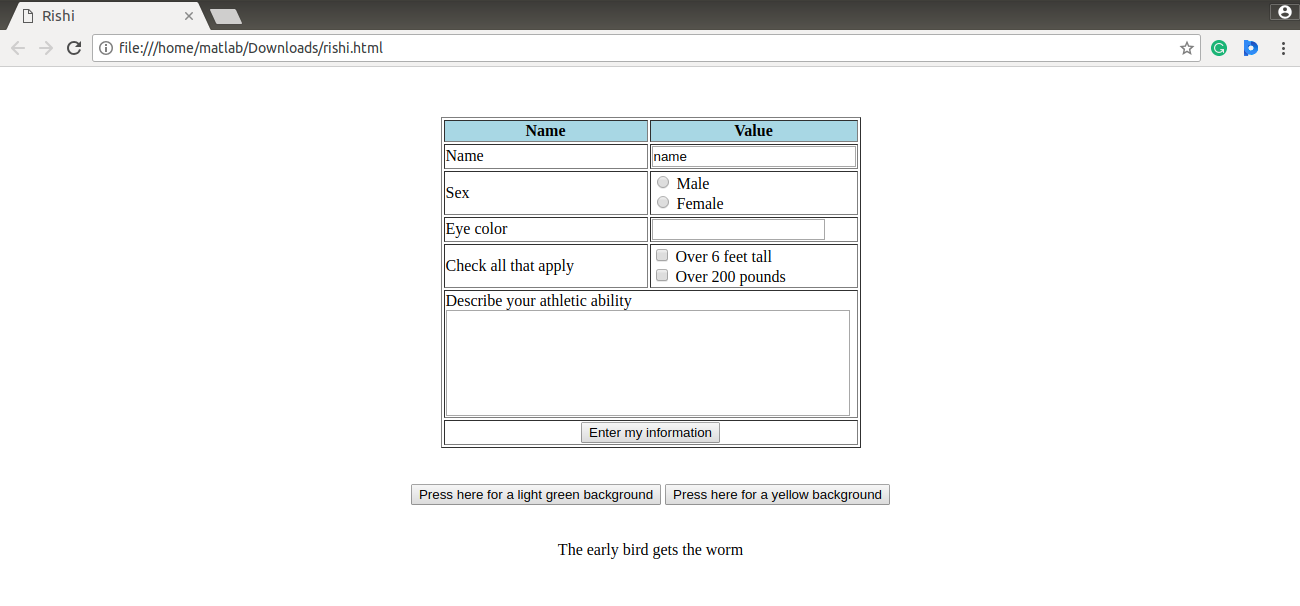
**EXPERIMENT 2**

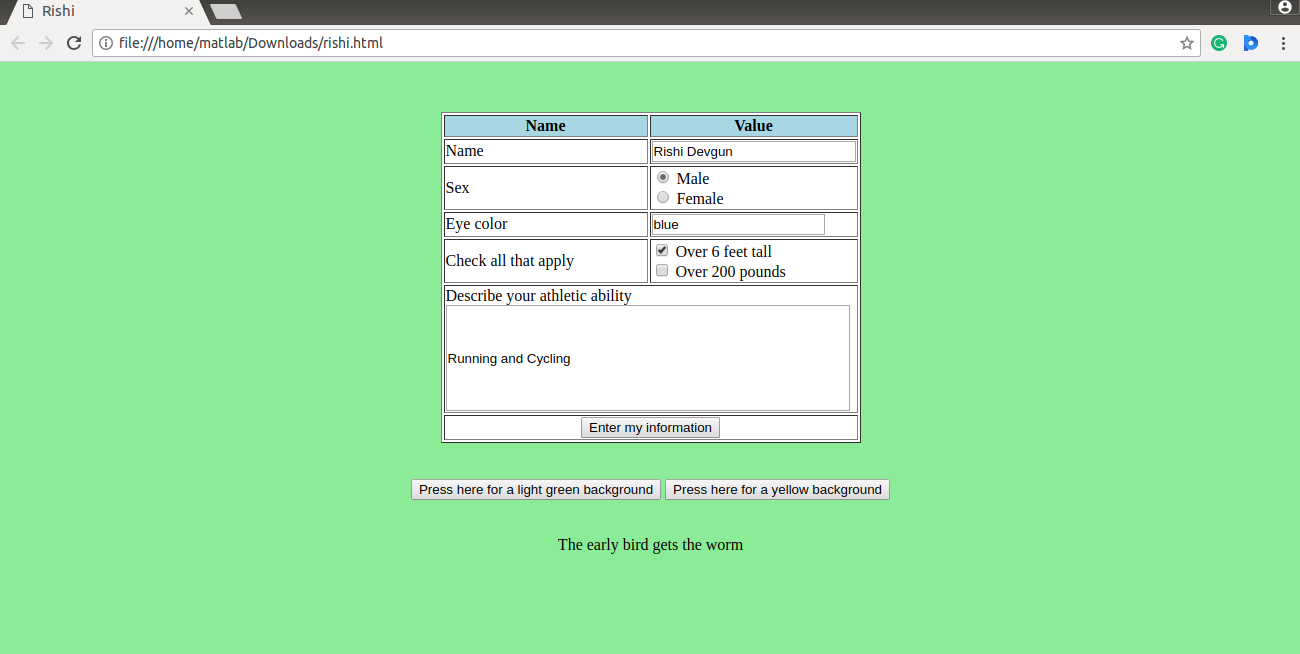
****

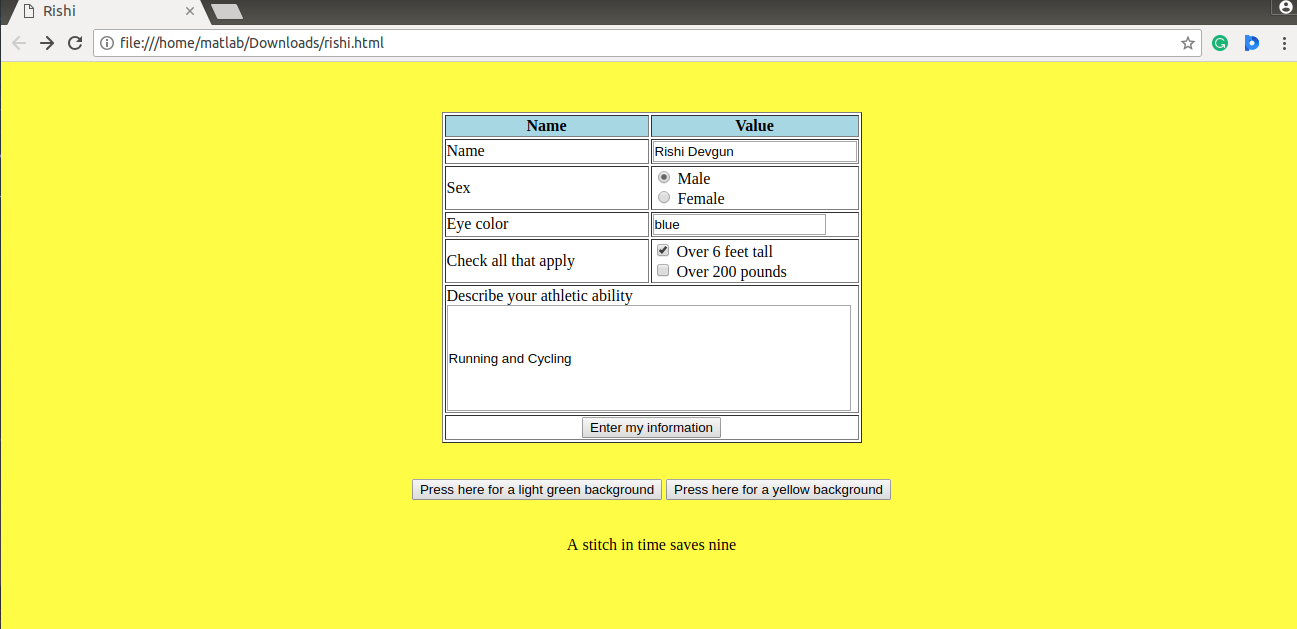
**Code:**

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <style type="text/css"> |
|  |  |
|  | form{ |
|  | padding: 10px; |
|  | } |
|  | h4{ |
|  | text-align: center; |
|  | font-size: 20px; |
|  | margin-bottom: 5px; |
|  | } |
|  | textarea{ |
|  | margin-top: 20px; |
|  | } |
|  | input, select{ |
|  | width:100%; |
|  | border: 2px solid green; |
|  | padding: 5px; |
|  | float: left; |
|  | resize:vertical; |
|  | } |
|  | input[type="submit"]{ |
|  | background-color: darkgreen; |
|  | color: white; |
|  | margin:20px; |
|  | width:50%; |
|  | font-size: 16px; |
|  | float: none; |
|  | border: 2px solid grey; |
|  | border-radius: 5px; |
|  | } |
|  | label{ |
|  | float: left; |
|  | border: 2px solid green; |
|  | text-align: right; |
|  | padding-right: 5px; |
|  | margin-right: 10px; |
|  | width: 100%; |
|  | font-size: 12pt; |
|  | padding-bottom: 1px; |
|  | padding-top: 1px |
|  | } |
|  | .Box{ |
|  | border: 2px solid green ; |
|  | border-radius: 15px; |
|  | background-color: #80dfff; |
|  | height: 550px; |
|  | margin: 20px 300px 20px 300px; |
|  | text-align: center; |
|  | } |
|  | .Index{ |
|  | width:25%; |
|  | float: left; |
|  | margin-top: 15px; |
|  | } |
|  | .Entry{ |
|  | width:70%; |
|  | float: left; |
|  | margin-top: 10px; |
|  | margin-left: 15px; |
|  | } |
|  | .row:after { |
|  | content: ""; |
|  | display: table; |
|  | clear: both; |
|  | } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <div class="Box"> |
|  | <h4>Course Information</h4> |
|  | <form> |
|  | <div class="row"></div> |
|  | <div class="Index"> |
|  | <label>Name</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="text" name="name"> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Email</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="email" name="email"> |
|  | </div> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Gender</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <select> |
|  | <option>Male</option> |
|  | <option>Female</option> |
|  | <option>Others</option> |
|  | </select> |
|  | </div> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Date of Birth</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="Date" name="dob"> |
|  | </div> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Course Name</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="text" name="CourseName"> |
|  | </div> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Course Code</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="text" name="CourseCode"> |
|  | </div> |
|  | </div> |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Slot</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <select> |
|  | <option>L1+L2</option> |
|  | <option>L3+L4</option> |
|  | <option>L5+L6</option> |
|  | <option>L7+L8</option> |
|  | <option>L9+L10</option> |
|  | <option>L11+L12</option> |
|  | <option>L13+L14</option> |
|  | <option>L15+L16</option> |
|  | <option>L17+L18</option> |
|  | <option>L19+L20</option> |
|  | <option>L21+L22</option> |
|  | <option>L23+L24</option> |
|  | <option>L25+L26</option> |
|  | <option>L27+L28</option> |
|  | <option>L29+L30</option> |
|  | <option>L31+L32</option> |
|  | <option>L33+L34</option> |
|  | <option>L35+L36</option> |
|  | <option>L37+L38</option> |
|  | <option>L39+L40</option> |
|  | <option>L41+L42</option> |
|  | <option>L43+L44</option> |
|  | <option>L45+L46</option> |
|  | <option>L47+L48</option> |
|  | <option>L49+L50</option> |
|  | <option>L51+L52</option> |
|  | <option>L53+L54</option> |
|  | <option>L55+L56</option> |
|  | <option>L57+L58</option> |
|  | <option>L59+L60</option> |
|  | </select> |
|  | </div> |
|  | </div> |
|  |  |
|  | <div class="row"> |
|  | <div class="Index"> |
|  | <label>Mobile Number</label> |
|  | </div> |
|  | <div class="Entry"> |
|  | <input type="text" name="Mobile"> |
|  | </div> |
|  | </div> |
|  |  |
|  |  |
|  | <textarea rows="3" cols="80" name="remarks" placeholder="Remarks"></textarea> |
|  |  |
|  | <input type="submit" name="Submit"> |
|  | </form> |
|  | </div> |
|  | </body> |
|  | </html> |
|  |  |

**EXPERIMENT 3**

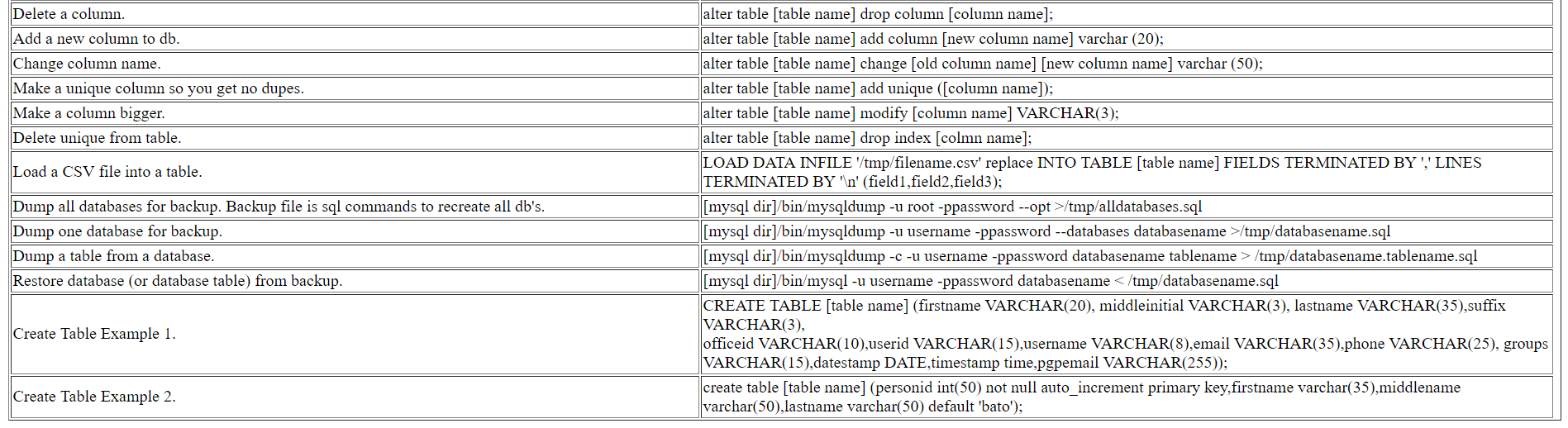
****

****

****

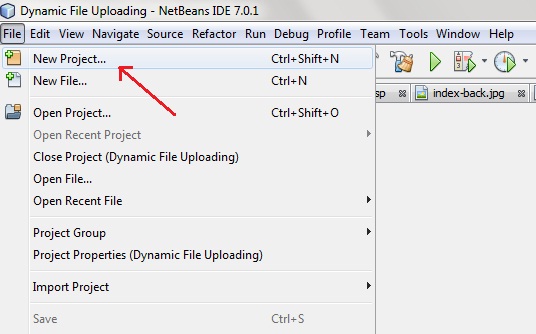
**Code:**

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <title> Rishi </title> |
|  | <style> |
|  |  |
|  | th { |
|  | width:200px; |
|  |  |
|  | } |
|  |  |
|  | input#text[type="text"] { |
|  | width: 400px; |
|  | height: 100px; |
|  | } |
|  |  |
|  | input[type="text"] { |
|  |  |
|  | width:200px; |
|  | } |
|  |  |
|  | form { |
|  |  |
|  | margin-top: 50px; |
|  | background-color: white; |
|  | width: 420px; |
|  |  |
|  | } |
|  |  |
|  | th { |
|  |  |
|  | background-color: lightblue; |
|  | } |
|  |  |
|  | </style> |
|  |  |
|  | <script> |
|  |  |
|  | function changeColor(color) |
|  |  |
|  | { |
|  | document.body.style.backgroundColor = color; |
|  |  |
|  | } |
|  |  |
|  |  |
|  | </script> |
|  |  |
|  | </head> |
|  | <body> |
|  | <center> |
|  | <div id="big"> |
|  | <form action="/action.php"> |
|  | <table border = 1> |
|  | <tr> |
|  | <th> Name </th> |
|  | <th> Value </th> |
|  | </tr> |
|  |  |
|  | <tr> |
|  | <td> Name </td> |
|  | <td> <input type="text" value="name" required> </td> |
|  | </tr> |
|  |  |
|  | <tr> |
|  | <td> Sex </td> |
|  | <td> <input type = "radio" value = "Male" name="gender" required> Male <br> |
|  | <input type = "radio" value = "Female" name="gender"> Female |
|  | </td> |
|  | </tr> |
|  |  |
|  | <tr> |
|  | <td> Eye color </td> |
|  | <td> <input list="color"><datalist id="color"> |
|  | <option value="green" required> |
|  | <option value="blue"> |
|  | <option value="black"> |
|  | </datalist> |
|  |  |
|  | </td> |
|  | </tr> |
|  |  |
|  | <tr> |
|  | <td> Check all that apply </td> |
|  | <td> <input type = "checkbox" value = "over 6ft" name="check" required> Over 6 feet tall <br> |
|  | <input type = "checkbox" value = "over 200" name="check"> Over 200 pounds </td> |
|  | </tr> |
|  |  |
|  | <tr> |
|  | <td colspan="2">Describe your athletic ability |
|  | <br><input type = "text" id="text" required> </td> |
|  | </tr> |
|  |  |
|  | <tr ><td colspan="2"><center><input type="submit" value="Enter my information"></center></td></tr> |
|  |  |
|  | </table> |
|  | </form> |
|  | <br><br> |
|  |  |
|  | <div id="buttons"> |
|  |  |
|  | <input type="submit" id="greenb" value="Press here for a light green background" onclick=changeColor("lightgreen")><span> |
|  | <input type="submit" id="yellowb" value="Press here for a yellow background" onclick=changeColor("yellow")></div><br><br> |
|  |  |
|  | <div id="quote"></div> |
|  |  |
|  | <script> |
|  | var saying=["The early bird gets the worm", "A stitch in time saves nine", "Make hay while the sun shines"]; |
|  | var i; |
|  |  |
|  | i = Math.floor(Math.random()\*4) |
|  | document.getElementById("quote").innerHTML=saying[i];</script> |
|  |  |
|  |  |
|  |  |
|  | </center> |
|  |  |
|  |  |
|  | </body> |
|  | </html> |

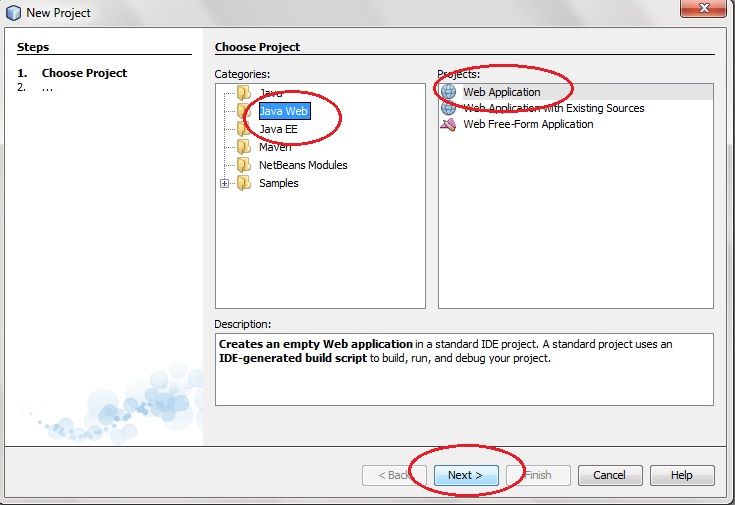


To create a servlet application in Netbeans IDE, you will need to follow the following (simple) steps :

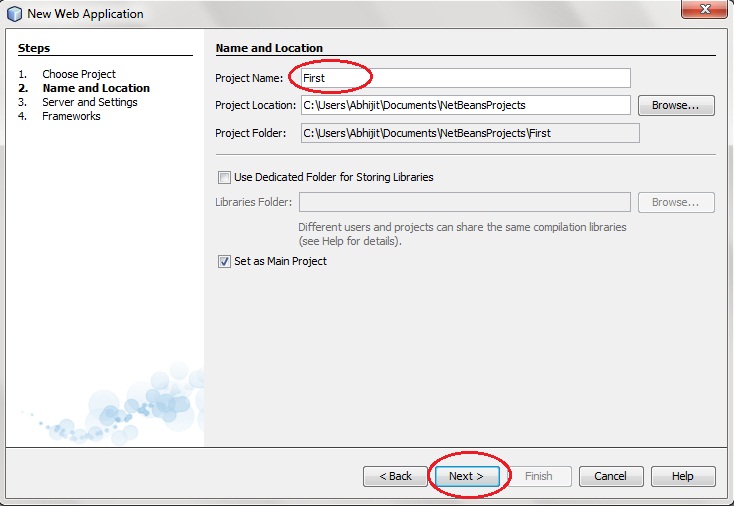
1. Open Netbeans IDE, Select **File** -> **New Project**



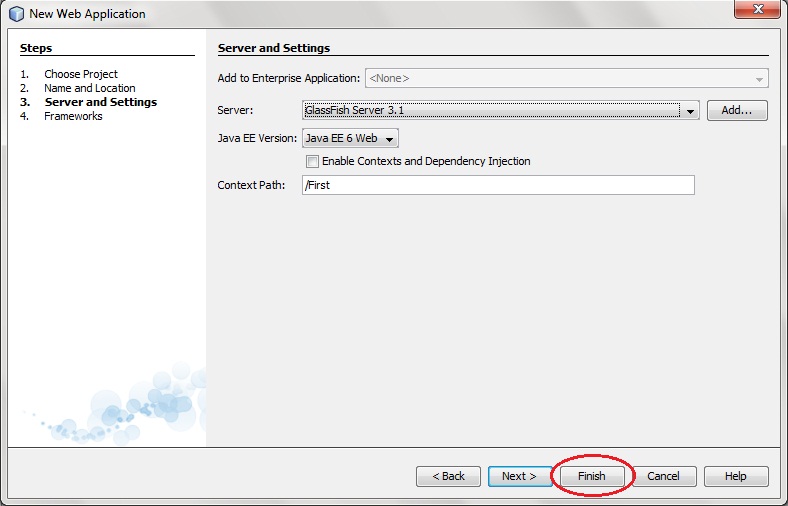
1. Select **Java Web** -> **Web Application**, then click on Next,



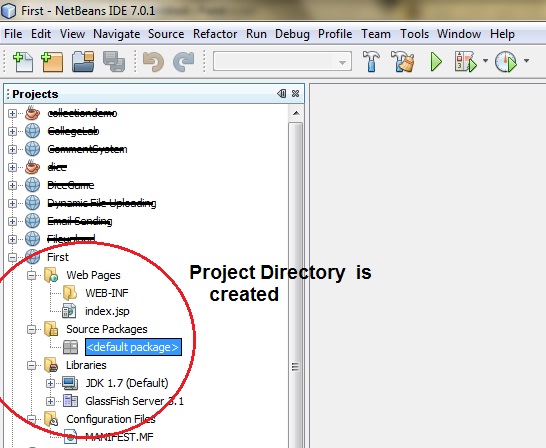
1. Give a name to your project and click on Next,



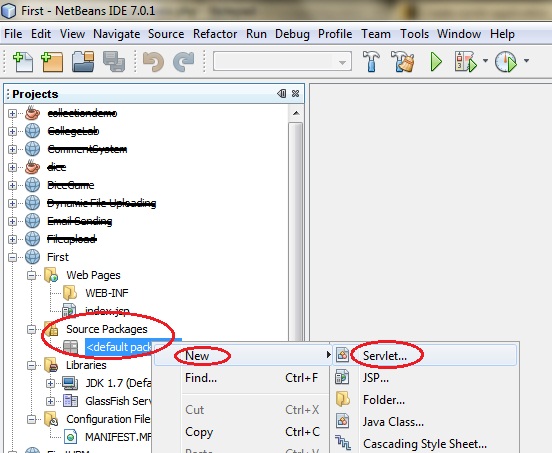
1. and then, Click **Finish**



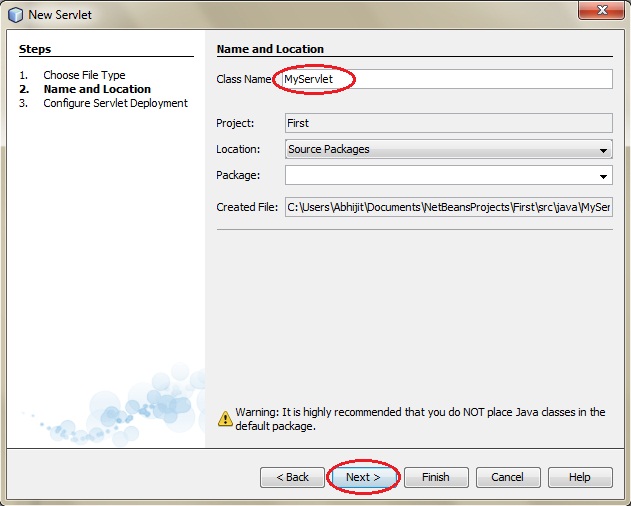
1. The complete directory structure required for the Servlet Application will be created automatically by the IDE.

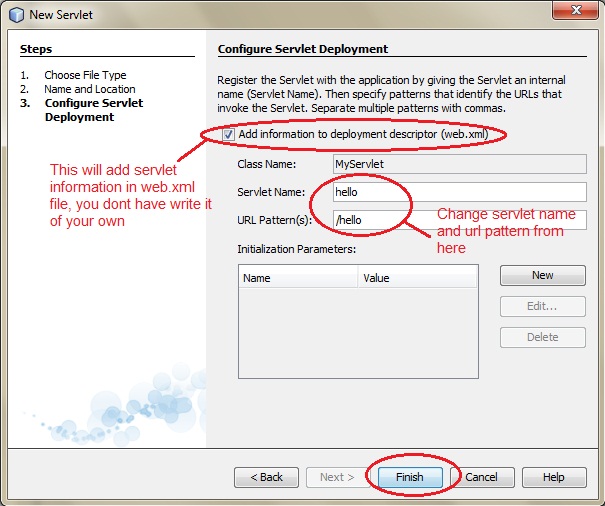


1. To create a Servlet, open **Source Package**, right click on **default packages** -> **New** -> **Servlet**.

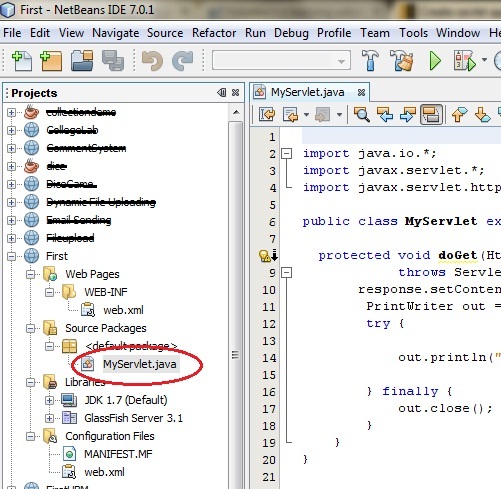


1. Give a Name to your Servlet class file,

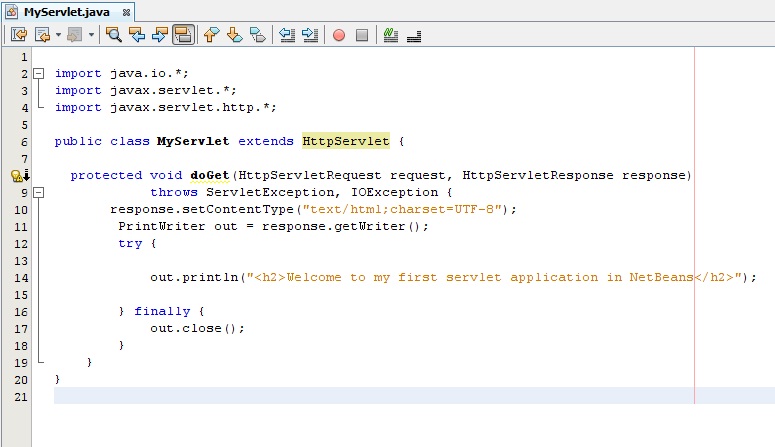




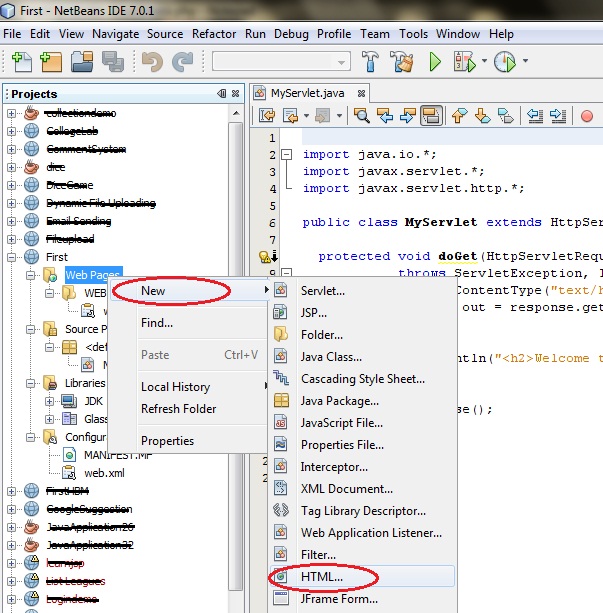
1. Now, your Servlet class is ready, and you just need to change the method definitions and you will good to go.



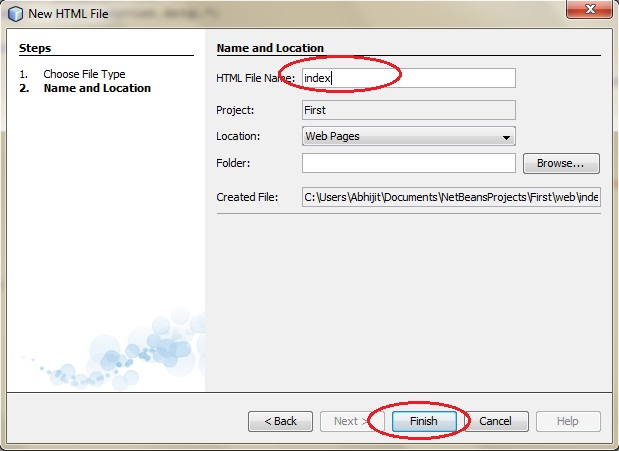
1. Write some code inside your Servlet class.



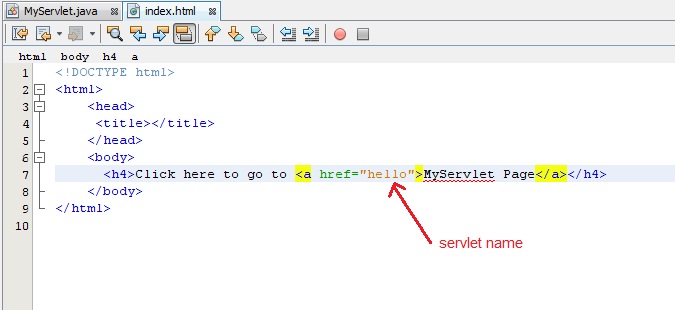
1. Create an HTML file, right click on **Web Pages** -> **New** -> **HTML**



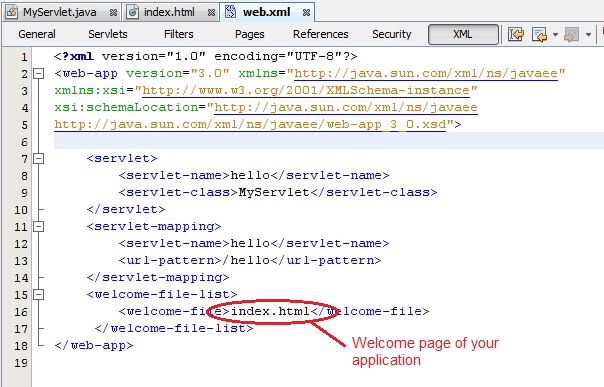
1. Give it a name. We recommend you to name it index, because browser will always pick up the index.html file automatically from a directory. Index file is read as the first page of the web application.



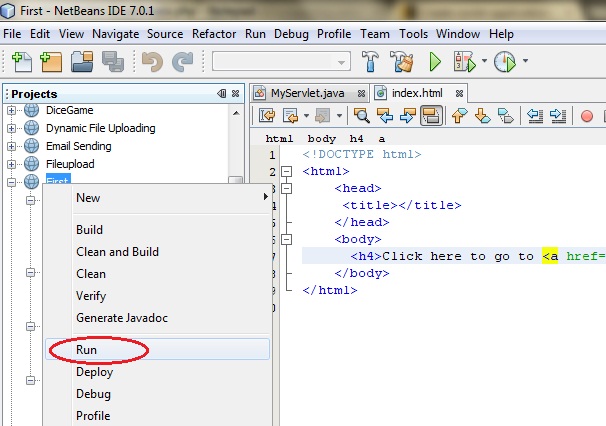
1. Write some code inside your HTML file. We have created a hyperlink to our Servlet in our HTML file.



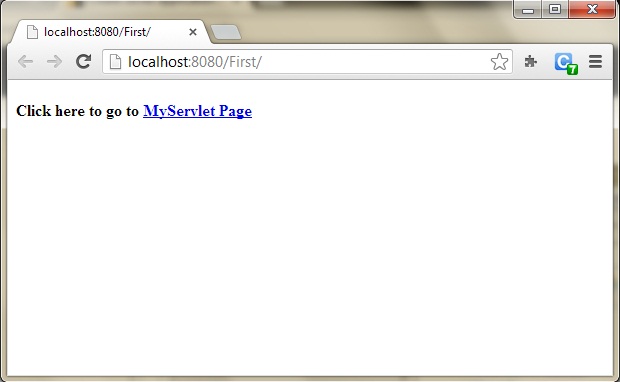
1. Edit **web.xml** file. In the web.xml file you can see, we have specified the **url-pattern** and the **servlet-name**, this means when hello url is accessed our Servlet file will be executed.



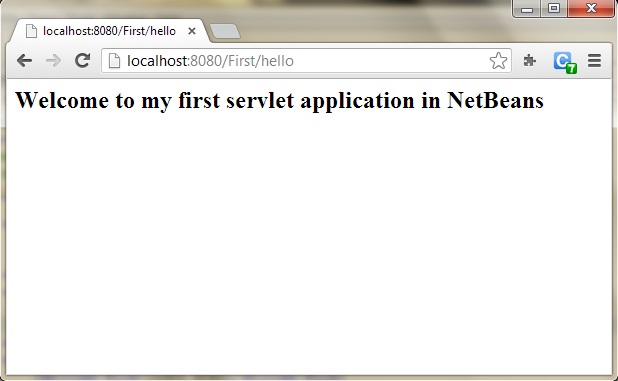
1. Run your application, right click on your Project and select **Run**



1. Click on the link created, to open your Servlet.



1. Hurray! Our First Servlet class is running.



**SERVELET:**

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

public class ProcessGuestBookServlet extends HttpServlet {

public void init(ServletConfig config)

throws ServletException {

super.init(config);

} // end init()

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String errMsg = "";

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

String surname = request.getParameter("surname");

String gender = request.getParameter("gender");

String occupation = request.getParameter("occupation");

String company = request.getParameter("company");

String emailAddress = request.getParameter("emailAddress");

String websiteAddress = request.getParameter("websiteAddress");

String homeAddress = request.getParameter("homeAddress");

String postCode = request.getParameter("postCode");

String city = request.getParameter("city");

String country = request.getParameter("country");

String comment = request.getParameter("comment");

// We will use this one to check if the e-mail address does nopt start with @

int emailAddressStartsWithAt = emailAddress.indexOf('@');

// Now let's do some validation

if(name.length() == 0) {

errMsg += "<br>Name has not been supplied!<br>";

} else if(surname.length() == 0) {

errMsg += "Surname not supplied!<br>";

} else if(gender == null) {

errMsg += "You did not tell us what gender you are!<br>";

} else if(country.equals("none")) {

errMsg += "Please choose your country from the drop down list<br>";

} else if(emailAddress.length() == 0) {

errMsg += "E-mail Address was not provided";

} else if(emailAddress.indexOf('@') <= -1) {

errMsg += "Invalid e-mail address - @ is not present";

} else if(emailAddressStartsWithAt == 0) {

errMsg += "<br>Invalid e-mail Address! It starts with @!!<br>";

} else {

// Process the data. Here you can insert into a db or whatever!

} // end validation

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<html><head>");

out.println("<title> Process GuestBook Servlet </title>");

out.println("<head><body topmargin=0 leftmargin=0>");

if(errMsg.length() != 0) {

out.println("Sorry! there were errors. Error message is " + errMsg);

} else {

out.println("Congratulations! No errors were found!");

} // end if

out.println("</body></html>");

} // end doPost() } // end ProcessGuestBookServlet

SERVELETS EX:

1)

import java.io.\*;

import javax.servlet.\*;

public class First implements Servlet {

ServletConfig config = null;

public void init(ServletConfig config) {

this.config = config;

System.out.println("servlet is initialized");

}

public void service(ServletRequest req, ServletResponse res)

throws IOException, ServletException {

res.setContentType("text/html");

PrintWriter out = res.getWriter();

out.print("<html><body>");

out.print("<b>hello simple servlet</b>");

out.print("</body></html>");

}

public void destroy() { System.out.println("servlet is destroyed"); }

public ServletConfig getServletConfig() { return config; }

public String getServletInfo() { return "copyright 2007-1010"; }

}

2)

/\* Generic Servlet \*/

import java.io.\*;

import javax.servlet.\*;

public class First extends GenericServlet {

public void service(ServletRequest req, ServletResponse res)

throws IOException, ServletException {

res.setContentType("text/html");

PrintWriter out = res.getWriter();

out.print("<html><body>");

out.print("<b>hello generic servlet</b>");

out.print("</body></html>");

}}

3)

/\* Servlet Lifecycle \*/

// init method is invoked

public void init(ServletConfig config) throws ServletException

// service method is invoked

public void service(ServletRequest request, ServletResponse response)

throws ServletException, IOException

// destroy method is invoked

public void destroy()

4)

/\* Servlet example \*/

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

res.setContentType("text/html"); // setting the content type

PrintWriter pw = res.getWriter(); // get the stream to write the data

// writing html in the stream

pw.println("<html><body>");

pw.println("Welcome to servlet");

pw.println("</body></html>");

pw.close(); // closing the stream

}}

5)

/\* Inside the public service method \*/

public void service(ServletRequest req, ServletResponse res)

throws ServletException, IOException {

HttpServletRequest request;

HttpServletResponse response;

try {

request = (HttpServletRequest)req;

response = (HttpServletResponse)res;

}

catch(ClassCastException e) {

throw new ServletException("non-HTTP request or response");

}

service(request, response);

}

/\* Inside the protected service method \*/

protected void service(HttpServletRequest req, HttpServletResponse resp)

throws ServletException, IOException {

String method = req.getMethod();

if(method.equals("GET")) {

long lastModified = getLastModified(req);

if(lastModified == -1L) { doGet(req, resp); }

...

// rest of the code

}

}

6)

/\* ServletRequest interface \*/

// index.html

<form action = "welcome" method = "get">

Enter your name <input type = "text" name = "name"><br>

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

</form>

// DemoServ.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

public class DemoServ extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

res.setContentType("text/html");

PrintWriter pw = res.getWriter();

String name = req.getParameter("name"); // will return value

pw.println("Welcome" + name);

pw.close();

}}

7)

/\* Displaying all header info in Servlet \*/

// Syntax of getHeaderNames() method

public Enumeration getHeaderNames()

// Syntax of getHeader() method

public String getHeader(String headerName)

// index.html

<a href = "run"> click here </a>

// ShowHeaders.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class ShowHeaders extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws IOException, ServletException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("HTTP headers sent by your client: <br>");

Enumeration enum = request.getHeaderNames();

while(enum.hasMoreElements()) {

String headerName = (String) enum.nextElement();

String headerValue = request.getHeader(headerName);

out.print("<b>" + headerName + "</b>: ");

out.println(headerValue + "<br>");

}

}}

8)

/\* RequestDispatcher in Servlet \*/

// Syntax of getRequestDispatcher method

public RequestDispatcher getRequestDispatcher (String resource);

// Example of using getRequestDispatcher method

RequestDispatcher rd = request.getRequestDispatcher("servlet2");

// servlet2 is the url-pattern of the second servlet

rd.forward(request, response); // Method may be include or forward

// index.html

<form action = "servlet1" method = "post">

Name: <input type = "text" name = "userName"/><br/>

Password: <input type = "password" name = "userPass"/><br/>

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

</form>

// Login.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class Login extends HttpServlet {

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

String p = request.getParameter("userPass");

if(p.equals("servlet") {

RequestDispatcher rd = request.getRequestDispatcher("servlet2");

rd.forward(request, response);

}

else {

out.print("Sorry UserName or Password error!");

RequestDispatcher rd = request.getRequestDispatcher("/index.html");

rd.include(request, response);

}

}}

// WelcomeServlet.java

import.java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class WelcomeServlet extends HttpServlet {

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

out.print("Welcome" + n);

}}

9)

/\* sendRedirect in Servlet \*/

// Syntax of sendRedirect() method

public void sendRedirect(String URL) throws IOException;

// example of sendRedirect() method

response.sendRedirect("http://www.javatpoint.com");

// DemoServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest req,

HttpServletResponse res)

throws ServletException, IOException {

res.setContentType("text/html");

PrintWriter pw = res.getWriter();

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

pw.close();

}}

/\* Creating custom google search using sendRedirect \*/

// index.html

<!DOCTYPE html>

<html>

<head>

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

<title>sendRedirect example</title>

</head>

<body>

<form action = "MySearcher">

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

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

</form>

</body>

</html>

// MySearcher.java

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class MySearcher extends HttpServlet {

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

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

response.sendRedirect("https://www.google.co.in/#q=" + name);

}}

10)

/\* ServletConfig interface \*/

// Syntax of getServletConfig() method

public ServletConfig getServletConfig();

// Example of getServletConfig() method

ServletConfig config = getServletConfig();

// Now we can call the methods of ServletConfig interface

// Example of ServletConfig to get 1 initialization parameter

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

ServletConfig config = getServletConfig();

String driver = config.getInitParameter("driver");

out.print("Driver is: " + driver);

out.close();

}}

// Example of ServletConfig to get all initialization parameters

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Enumeration;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

ServletConfig config = getServletConfig();

Enumeration<String> e= config.getInitParamterNames();

String str = "";

while(e.hasMoreElements()) {

str = e.nextElement();

out.print("<br>Name: " + str);

out.print(" value: " + config.getInitParameter(str));

}

out.close();

}}

11)

/\* ServletContext interface \*/

// Syntax of getServletContext() method

public ServletContext getServletContext()

// Example of getServletContext() method

// We can get the ServletContext object from ServletConfig object

ServletContext application = getServletConfig().getServletContext();

// Another convenient way to get the ServletContext object

ServletContext application = getServletContext();

// Example of ServletContext to get 1 initialization parameter

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest req,

HttpServletResponse res)

throws ServletException, IOException {

res.setContentType("text/html");

PrintWriter pw = res.getWriter();

// Creating ServletContext object

ServletContext context = getServletContext();

// Getting the value of the initialization parameter & printing it

String driverName = context.getInitParameter("dname");

pw.println("driver name is = " + driverName);

pw.close();

}}

// Example of ServletContext to get all initialization parameters

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet extends HttpServlet {

public void doGet(HttpServletRequest req,

HttpServletResponse res)

throws ServletException, IOException {

res.setContentType("text/html");

PrintWriter out = res.getWriter();

ServletContext context = getServletContext();

Enumeration<String> e = context.getInitParameterNames();

String str = "";

while(e.hasMoreElements()) {

str = e.nextElement();

out.print("<br>" + context.getInitParameter(str));

}

}}

12)

/\* Attribute in Servlet \*/

// DemoServlet1.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet1 extends HttpServlet {

public void doGet(HttpServletRequest req,

HttpServletResponse res) {

try {

res.setContentType("text/html");

PrintWriter out = res.getWriter();

ServletContext context = getServletContext();

context.setAttribute("company", "IBM");

out.println("Welcome to first servlet");

out.println("<a href = 'servlet2'>visit</a>");

out.close();

} catch(Exception e) { out.println(e); }

}}

// DemoServlet2.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DemoServlet2 extends HttpServlet {

public void doGet(HttpServletRequest req,

HttpServletResponse res) {

try {

res.setContentType("text/html");

PrintWriter out = res.getWriter();

ServletContext context = getServletContext();

String n = (String)context.getAttribute("company");

out.println("Welcome to " + n);

out.close();

}catch(Exception e) { out.println(e); }

}}

13)

/\* Cookies in Servlet \*/

// Creating a cookie

Cookie ck = new Cookie("user", "ana"); // creating cookie object

response.addCookie(ck); // adding cookie in the response

// Deleting a cookie

Cookie ck = new Cookie("user", ""); // deleting value of cookie

ck.setMaxAge(0); // changing the maximum age to 0 seconds

response.addCookie(ck); // adding cookie in the response

// Getting all cookies

Cookie ck[] = request.getCookies();

for(int i = 0; i < ck.length; i++) {

out.print("<br>" + ck[i].getName() + " " + ck[i].getValue());

// printing name and value of cookie

}

// index.html

<form action = "servlet1" method = "post">

Name: <input type = "text" name = "userName"/><br/>

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

</form>

// FirstServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

public void doPost(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

out.print("Welcome" + n);

Cookie ck = new Cookie("uname", n); // Creating cookie object

response.addCookie(ck); // Adding cookie in the response

// Creating submit button

out.print("<form action = 'servlet2'>");

out.print("<input type = 'submit' value = 'go'>");

out.print("</form>");

out.close();

}catch(Exception e) { System.out.println(e); }

}}

// SecondServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SecondServlet extends HttpServlet {

public void doPost(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

Cookie ck[] = request.getCookies();

out.print("Hello " + ck[0].getValue());

out.close();

}catch(Exception e) { System.out.println(e); }

}}

14)

/\* Servlet Login & Logout example using cookies \*/

// index.html

<!DOCTYPE html>

<html>

<head>

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

<title>Servlet Login Example</title>

</head>

<body>

<h1>Welcome to login app by cookie</h1>

<a href = "login.html">Login</a>

<a href = "LogoutServlet">Logout</a>

<a href = "ProfileServlet">Profile</a>

</body>

</html>

// link.html

<a href = "login.html">Login</a>

<a href = "LogoutServlet">Logout</a>

<a href = "ProfileServlet">Profile</a>

<hr>

// login.html

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

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

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

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

</form>

// LoginServlet.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

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

String password = request.getParameter("password");

if(password.equals("admin123")) {

out.print("You are successfully logged in");

out.print("<br>Welcome, " + name);

Cookie ck = new Cookie("name", name);

response.addCookie(ck);

}else{

out.print("sorry, username or password error!");

request.getRequestDispatcher("login.html").include(request, response);

}

out.close();

}}

// LogoutServlet.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class LogoutServlet extends HttpServlet {

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

Cookie ck = new Cookie("name", "");

ck.setMaxAge(0);

response.addCookie(ck);

out.print("You are successfully logged out");

}}

// ProfileServlet.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class ProfileServlet extends HttpServlet {

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

Cookie ck[] = request.getCookies();

if(ck != null) {

String name = ck[0].getValue();

if(!name.equals("") || name != null) {

out.print("<b>Welcome to Profile</b>");

out.print("<br>Welcome, " + name);

}

}else{

out.print("Please login first");

request.getRequestDispatcher("login.html").include(request, response);

}

out.close();

}}

15)

/\* Hidden Form Field using Servlets \*/

// index.html

<form action = "servlet1">

Name: <input type = "text" name = "userName"/><br/>

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

</form>

// FirstServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

out.print("Welcome " + n);

// Creating form that has invisible textfield

out.print("<form action = 'servlet2'>");

out.print("<input type = 'hidden' name = 'uname' value='" + n + "'>");

out.print("<input type = 'submit' value = 'go'>");

out.print("</form>");

out.close();

}catch(Exception e) { System.out.println(e); }

}}

// SecondServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SecondServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Getting the value from the hidden field

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

out.print("Hello " + n);

out.close();

}catch(Exception e) { System.out.println(e); }

}}

16)

/\* URL rewriting using Servlets \*/

// index.html

<form action = "servlet1">

Name: <input type = "text" name = "userName"/><br/>

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

</form>

// FirstServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

out.print("Welcome " + n);

// Appending the username in the query string

out.print("<a href = 'servlet2?uname=" + n + "'>visit</a>");

out.close();

}catch(Exception e) { System.out.println(e); }

}}

// SecondServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SecondServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Getting value from the query string

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

out.print("Hello " + n);

out.close();

}catch(Exception e) { System.out.println(e); }

}}

17)

/\* HttpSession interface in Servlets \*/

// index.html

<form action = "servlet1">

Name: <input type = "text" name = "userName"/><br/>

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

</form>

// FirstServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n = request.getParameter("userName");

out.print("Welcome " + n);

HttpSession session = request.getSession();

session.setAttribute("uname", n);

out.print("<a href = 'servlet2'>visit</a>");

out.close();

}catch(Exception e) { System.out.println(e); }

}}

// SecondServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SecondServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response) {

try {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession(false);

String n = (String)session.getAttribute("uname");

out.print("Hello " + n);

out.close();

}catch(Exception e) { System.out.println(e); }

}}

18)

/\* HttpSession Login & Logout using Servlets \*/

// index.html

<!DOCTYPE html>

<html>

<head>

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

<title>Servlet Login Example</title>

</head>

<body>

<h1>Login App using HttpSession</h1>

<a href = "login.html">Login</a>

<a href = "LogoutServlet">Logout</a>

<a href = "ProfileServlet">Profile</a>

</body>

</html>

// link.html

<a href = "login.html">Login</a>

<a href = "LogoutServlet">Logout</a>

<a href = "ProfileServlet">Profile</a>

<hr>

// LoginServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

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

String password = request.getParameter("password");

if(password.equals("admin123")) {

out.print("Welcome, " + name);

HttpSession session = request.getSession();

session.setAttribute("name", name);

}else{

out.print("Sorry, username or password error!");

request.getRequestDispatcher("login.html").include(request, response);

}

out.close();

}}

// LogoutServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class LogoutServlet extends HttpServlet {

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

HttpSession session = request.getSession();

session.invalidate();

out.print("You are successfully logged out");

out.close();

}}

// ProfileServlet.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class ProfileServlet extends HttpServlet {

protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

HttpSession session = request.getSession(false);

if(session != null) {

String name = (String)session.getAttribute("name");

out.print("Hello, " + name + " Welcome to Profile");

}else{

out.print("Please login first");

request.getRequestDispatcher("login.html").include(request, response);

}

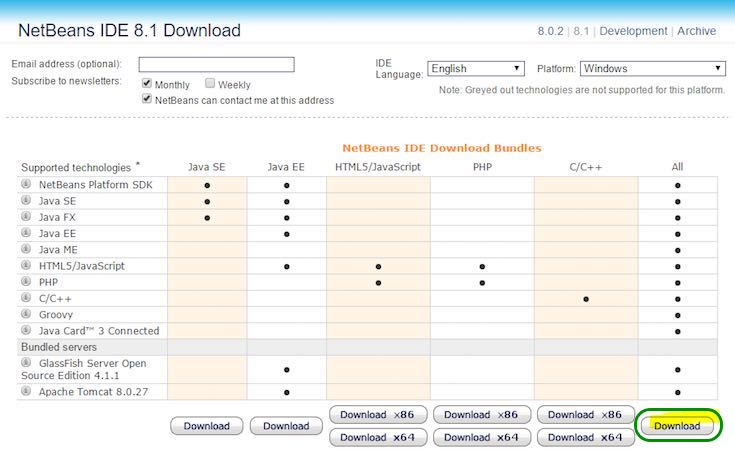
out.close();

}}

**JSP:**

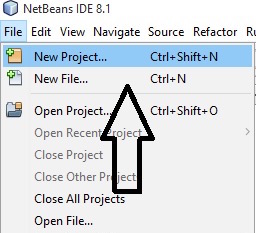
### Step1 −

Download NetBeans IDE from NetBeans official website and install it on your computer. On opening the link you will find the following. I have downloaded the yellow marked one as in the following screenshot. While installing add the JAVA path i.e. [download java](http://www.oracle.com/technetwork/java/javase/downloads/index.html) before installing NetBeans IDE.



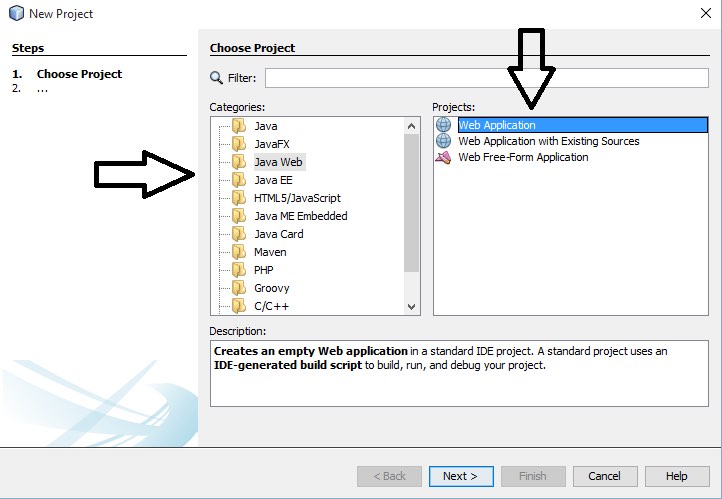
### Step 2 −

Open NetBeans IDE & click **File** menu and click **New Project** as in the following screenshot −



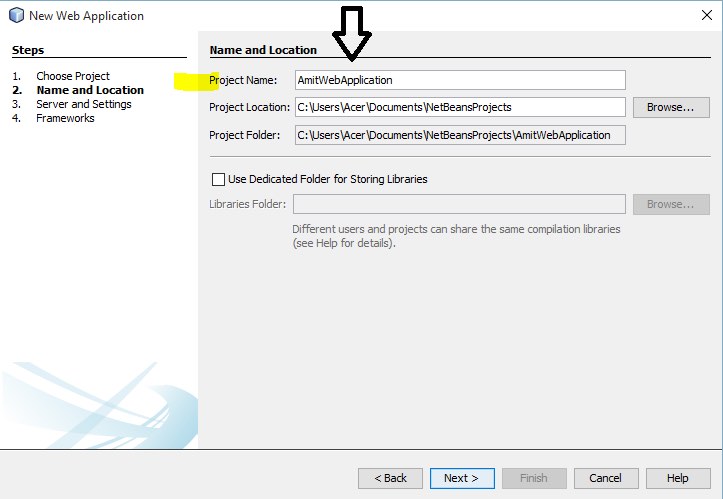
### Step 3 −

After clicking **New Project**, you will get the following dialog box. Select Java**Web**, after that **Web Application.**Now click **Next**to proceed further.

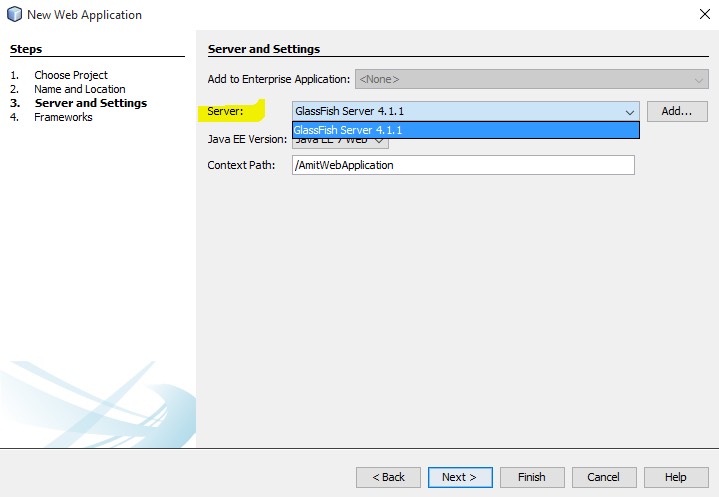


### Step 4 −

A new dialog box will open; now add project name and location. You can add any name; I have mentioned the project name as **AmitWebApplication** and kept the location as it is. After that click **Next**.

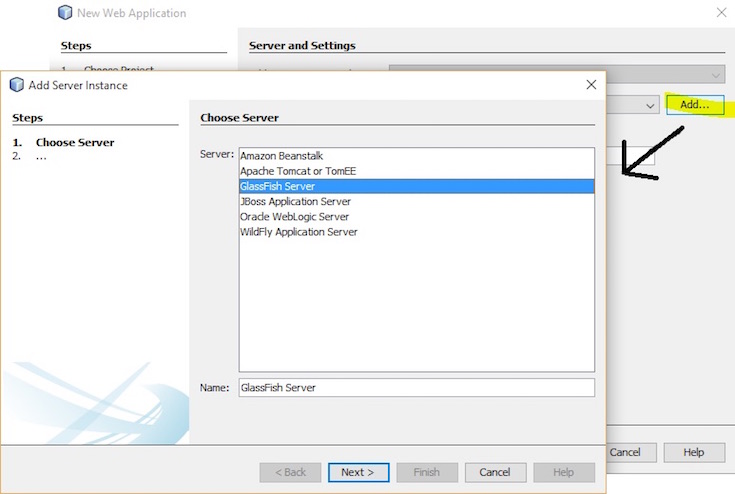


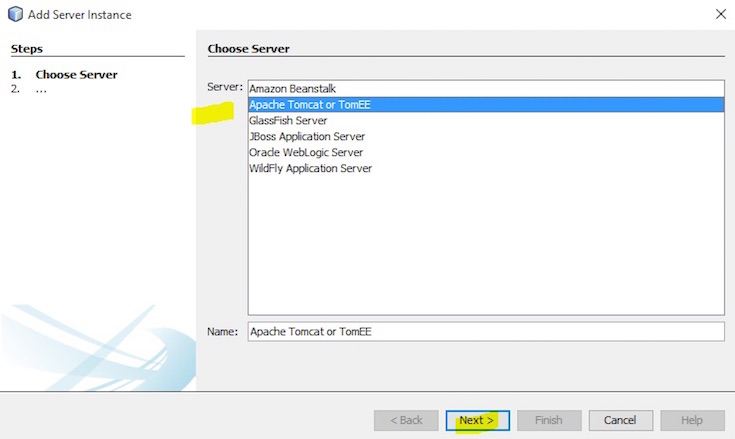
### Step 5 −

Now you need to select a server, b y default glassfish server is provided, but we will consider Apache Tomcat Server as mentioned before.  
  
For that got to download Apache Tomcat 7 page and download **64-bit Windows zip (pgp, md5, sha1)**. I am having a Windows 10 64-bit OS, so I downloaded the 64-bit version.

Place the file in C: drive (you don’t need installation because we did not download the installer), so your final path for Apache Tomcat on my system is −

C:\apache-tomcat-7.0.67

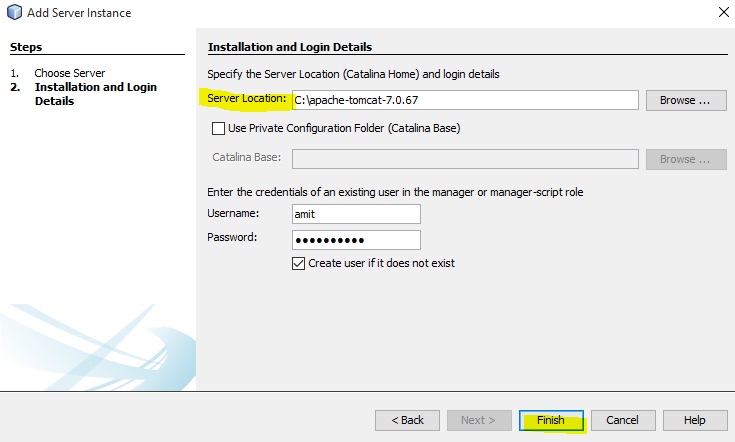
After installing return back to NetBeans IDE and click **Add** in the same dialog box as in the following screenshot. You will see a list of servers under **Add Server Instance** dialog box.  


Now select **Apache Tomcat**and click **Next** −  


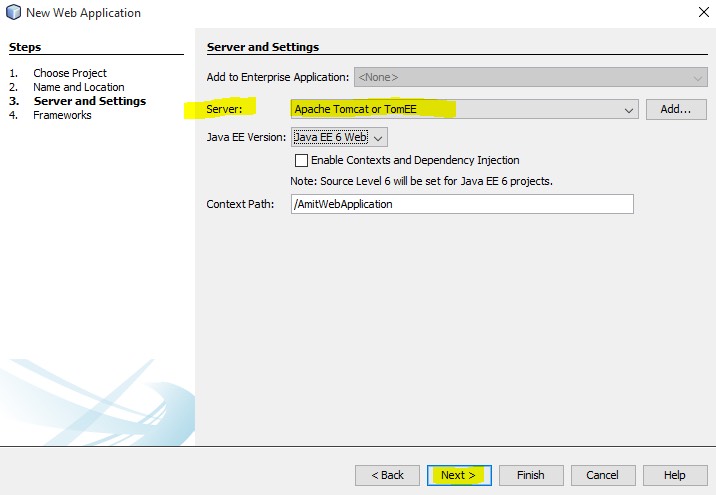
The new dialog box will ask for adding the server location. Add the path where you installed Apache Tomcat Server. The path is −

C:\apache-tomcat-7.0.67

Here’s the screenshot −



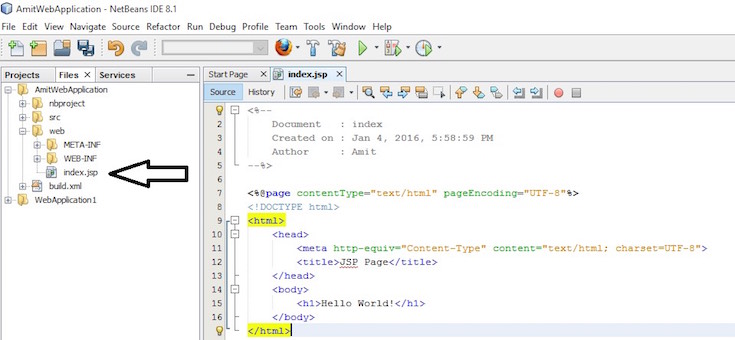
Now, click **Finish.**After clicking, you will return back to Add Server dialog with Server as Apache Tomcat as in the following screenshot −



### Step 6 −

After that click **Next** and do not select any Framework since we’re considering a simple program here. Click **Finish**.

### Step 7 −

Now you can see the following default JSP will open. Also, check the directory structure on the left in the following screenshot. You can see the file **index.jsp** under **web** folder −  


Now edit the JSP file; here’s the edited code. We will just add some text to print the date and time. For this, I have taken an example of [JSP Expressions](http://www.tutorialspoint.com/jsp/jsp_syntax.htm).

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

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

<title>First JSP Page</title>

</head>

<body>

<h1>Today's date</h1>

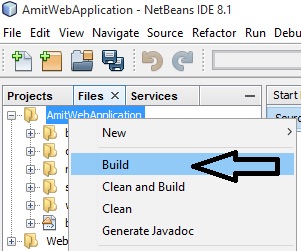
Date and Time: <%= (new java.util.Date().toString())%>

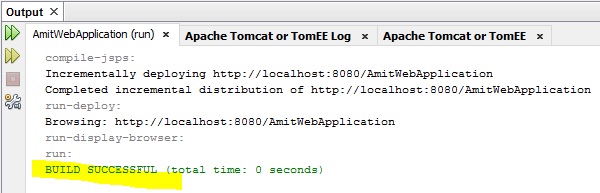
</body>

</html>

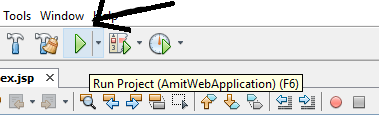
### Step 8 −

Now to run the**index.jsp** file, build the project as in the following screenshot. Right click on the project**“AmitWebApplication”** and click **Build** −



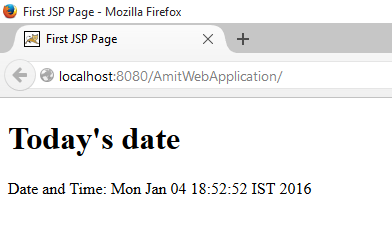
After that you can see that the project build works fine.  


Now, click F6 to run the project. You can also click on the following toolbar button −



Here’s the output if everything works fine. You can check the title, heading, and the date-time we printed using JSP Expressions. The project is executing on localhost, so the link is −

http://localhost:8080/AmitWebApplication/



## Scripting elements

A) scriptlet tag

Syntax:

<% java source code %>

Example:

<html>

<body>

<% out.print("welcome to jsp"); %>

</body>

</html>

--> Example of scriptlet tag that prints user name

index.html

<html>

<body>

<form action = "welcome.jsp">

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

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

</form>

</body>

</html>

welcome.jsp

<html>

<body>

<%

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

out.print("welcome" + name);

%>

</form>

</body>

</html>

B) expression tag

Syntax:

<%= statement %>

Example:

<html>

<body>

<%= "welcome to jsp" %>

</body>

</html>

--> Example of expression tag that prints current time

index.jsp

<html>

<body>

Current Time: <%= java.util.Calendar.getInstance().getTime() %>

</body>

</html>

--> Example of expression tag that prints user name

index.jsp

<html>

<body>

<form action = "welcome.jsp">

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

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

</form>

</body>

</html>

welcome.jsp

<html>

<body>

<%= "Welcome" + request.getParameter("uname") %>

</body>

</html>

C) declaration tag

Syntax:

<%! field or method declaration %>

--> Example of declaration tag that declares field

index.jsp

<html>

<body>

<%! int data = 50; %>

<%= "Value of the variable is: " + data %>

</body>

</html>

--> Example of declaration tag that declares method

index.jsp

<html>

<body>

<%!

int cube(int n) {

return n\*n\*n;

}

%>

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

</body>

</html>

------------------------------------------------------

## Implicit objects

A) Request

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);

%>

B) Response

index.html

<form action = "welcome.jsp">

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

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

</form>

welcome.jsp

<%

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

%>

C) Config

index.html

<form action = "welcome">

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

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

</form>

web.xml

<web-app>

<servlet>

<servlet-name>abc</servlet-name>

<jsp-file>/welcome.jsp</jsp-file>

<init-param>

<param-name>dname</param-name>

<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>abc</servlet-name>

<url-pattern>/welcome</url-pattern>

</servlet-mapping>

</web-app>

welcome.jsp

<%

out.print("Welcome " + request.getParameter("uname"));

String driver = config.getInitParameter("dname");

out.print("driver name is = " + driver);

%>

D) Application

index.html

<form action = "welcome">

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

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

</form>

web.xml

<web-app>

<servlet>

<servlet-name>abc</servlet-name>

<jsp-file>/welcome.jsp</jsp-file>

</servlet>

<servlet-mapping>

<servlet-name>abc</servlet-name>

<url-pattern>/welcome</url-pattern>

</servlet-mapping>

<context-param>

<param-name>dname</param-name>

<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>

</context-param>

</web-app>

welcome.jsp

<%

out.print("Welcome " + request.getParameter("uname"));

String driver = application.getInitParameter("dname");

out.print("driver name is = " + driver);

%>

E) Session

index.html

<html>

<body>

<form action = "welcome.jsp">

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

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

</form>

</body>

</html>

welcome.jsp

<html>

<body>

<%

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

out.print("Welcome " + name);

session.setAttribute("user", name);

<a href = "second.jsp">second jsp page</a>

%>

</body>

</html>

second.jsp

<html>

<body>

<%

String name = (String)session.getAttribute("user");

out.print("Hello " + name);

%>

</body>

</html>

F) pageContext

index.html

<html>

<body>

<form action = "welcome.jsp">

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

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

</form>

</body>

</html>

welcome.jsp

<html>

<body>

<%

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

out.print("Welcome" + name);

pageContext.setAttribute("user", name, PageContext.SESSION\_SCOPE);

<a href = "second.jsp">second jsp page</a>

%>

</body>

</html>

second.jsp

<html>

<body>

<%

String name = (String)pageContext.getAttribute("user", PageContext.SESSION\_SCOPE);

out.print("Hello " + name);

%>

</body>

</html>

G) page

Object page = this;

<% (HttpServlet)page.log("message"); %>

<% this.log("message"); %>

H) exception

error.jsp

<%@ page isErrorPage = "true" %>

<html>

<body>

Sorry following exception occurred: <%= exception %>

</body>

</html>

### 2.  First JSP Example - "Java inside HTML"

Let's begin with a simple JSP example. We shall use the webapp called "hello" that we have created in our earlier exercise. Use a programming text editor to enter the following HTML/JSP codes and save as "first.jsp" (the file type of ".jsp" is mandatory) in your webapp (web context) home directory (i.e., "webapps\hello".

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <html>  <head><title>First JSP</title></head>  <body>  <%  double num = Math.random();  if (num > 0.95) {  %>  <h2>You'll have a luck day!</h2><p>(<%= num %>)</p>  <%  } else {  %>  <h2>Well, life goes on ... </h2><p>(<%= num %>)</p>  <%  }  %>  <a href="<%= request.getRequestURI() %>"><h3>Try Again</h3></a>  </body>  </html> |

To execute the JSP script: Simply start your Tomcat server and use a browser to issue an URL to browse the JSP page (i.e., http://localhost:8080/hello/first.jsp).

From your browser, choose the "View Source" option to check the response message. It should be either of the followings depending on the random number generated.

<html>

<h2>You'll have a luck day!</h2>

<p>(0.987)</p>

<a href="first.jsp"><h3>Try Again</h3></a>

</html>

<html>

<h2> Well, life goes on ... </h2>

<p>(0.501)</p>

<a href="first.jsp"><h3>Try Again</h3></a>

</html>

It is important to note that the client is not able to "view" the original JSP script (otherwise, you may have security exposure), but merely the result generated by the script.

**Explanations**

1. A JSP script is a regular HTML page containing Java programs. Recall that JSP is "Java inside HTML" (whereas servlet is "HTML inside Java"). The Java statements are enclosed by <% ... %> (called JSP scriptlet) or <%= ... %>(called JSP expression).
2. JSP Scriptlet <% ... %> is used to include Java statements.
3. JSP Expression <%= ... %> is used to evaluate a single Java expression and display its result.
4. The method request.getRequestURI() is used to retrieve the URL of the current page. This is used in the anchor tag <a> for refreshing the page to obtain another random number.

##### Behind the Scene

When a JSP is first accessed, Tomcat converts the JSP into a servlet; compile the servlet, and execute the servlet. Check out the generated servlet for "first.jsp", and study the JSP-to-servlet conversion. Look under Tomcat's "work\Catalina\localhost\hello" for "first\_jsp.java".

The relevant part of the generated servlet is extracted as follows (with some simplifications):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | out.write("<html>\r\n ");  double num = Math.random();  if (num > 0.95) {  out.write("<h2>You will have a luck day!");  out.write("</h2><p>(");  out.print( num );  out.write(")</p>\r\n");  } else {  out.write("\r\n ");  out.write("<h2>Well, life goes on ... ");  out.write("</h2><p>(");  out.print( num );  out.write(")</p>\r\n ");  }  out.write("<a href=\"");  out.print( request.getRequestURI() );  out.write("\">");  out.write("<h3>Try Again</h3></a>\r\n");  out.write("</html>\r\n"); |

##### Explanation

1. The HTML statements are written out as part of the response via out.write(), as "it is".
2. The JSP scriptlets <% ... %> are kept, as "it is", in the converted servlet as the program logic.
3. The JSP expressions <%= ... %> are placed inside a out.print(). Hence, the expression will be evaluated, and the result of the evaluation written out as part of the response message.

Compare the JSP script and the internally generated servlet, you shall understand that servlet is "HTML inside Java", whereas JSP is "Java inside HTML".

Subsequent accesses to the same JSP will be much faster, because they will be re-directed to the converted and compiled servlet directly (no JSP-to-servlet conversion and servlet compilation needed again), unless the JSP has been modified.

### 3.  Revisit Java Servlets

A typical Java servlet (as shown below) contains three kinds of methods: init(), destroy(), and one or more service() methods such as doGet() and doPost(). init() runs when the servlet is loaded. destroy() runs when the servlet is unloaded. service() runs once per HTTP request. The service() methods takes two arguments: request and response, corresponding to the HTTP request and response messages respectively. A PrintWriter called outis created for writing out the response to the network.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | import java.io.\*;  import javax.servlet.\*;  import javax.servlet.http.\*;    public class ...Servlet extends HttpServlet {    // Runs when the servlet is loaded onto the server.  public void init() {  ......  }    // Runs on a thread whenever there is HTTP GET request  // Take 2 arguments, corresponding to HTTP request and response  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws IOException, ServletException {    // Set the MIME type for the response message  response.setContentType("text/html");  // Write to network  PrintWriter out = response.getWriter();    // Your servlet's logic here  out.println("<html>");  out.println( ...... );  out.println("</html>");  }    // Runs as a thread whenever there is HTTP POST request  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws IOException, ServletException {  // do the same thing as HTTP GET request  doGet(request, response);  }    // Runs when the servlet is unloaded from the server.  public void destroy() {  ......  }    // Other instance variables and methods  } |

Java servlet produces HTML codes by calling out.print() methods. You have to hardcode all the HTML tags (and cannot use any WYSIWYG web authoring tools). Any change to the web page's presentation (such as background color and font size) requires re-coding and re-compilation of servlet program. Servlet, in a nutshell, is "HTML inside Java", whereas JSP is "Java inside HTML".

### 4.  Second JSP example - Echoing HTML Request Parameters

Enter the following JSP script and save as "echo.jsp" in your webapp's root directory.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | <html>  <head>  <title>Echoing HTML Request Parameters</title>  </head>  <body>  <h3>Choose an author:</h3>  <form method="get">  <input type="checkbox" name="author" value="Tan Ah Teck">Tan  <input type="checkbox" name="author" value="Mohd Ali">Ali  <input type="checkbox" name="author" value="Kumar">Kumar  <input type="submit" value="Query">  </form>    <%  String[] authors = request.getParameterValues("author");  if (authors != null) {  %>  <h3>You have selected author(s):</h3>  <ul>  <%  for (int i = 0; i < authors.length; ++i) {  %>  <li><%= authors[i] %></li>  <%  }  %>  </ul>  <a href="<%= request.getRequestURI() %>">BACK</a>  <%  }  %>  </body>  </html> |

Browse the JSP page created and study the generated servlet.

**Explanations**

1. This HTML page has a form with 3 checkboxes. The "name=value" pair of the checkboxes is "author=so\_and\_so". No "action" attribute is specified, the default "action" is the current page (i.e. the query will be sent to the same page).
2. The JSP scriptlet checks if the query parameter "author" exists to decide whether to dynamically generate the enclosed codes. "author" parameter is absent when the page is first requested. Once the client fills in the form (by checking the boxes) and submits the form, "author" will be present in the HTTP request, and submitted to the same page for processing (with the default <form>’s "action" attribute).
3. The request.getParameterValues() is used to retrieve all the values of the query parameter. The values are echoed back using an unordered list.

### 5.  JSP Scripting Elements

JSP provides the following scripting elements:

* JSP Comment <%-- comments -->
* JSP Expression <%= Java Expression %>
* JSP Scriptlet <% Java Statement(s) %>
* JSP Directive <%@ page|include ... %>

To simplify the access of the HTTP request and response messages, JSP has pre-defined the following variables:

* request: corresponds to the HTTP request message.
* response: corresponds to the HTTP response message.
* out: corresponds to the HTTP response message’s output stream.
* others such as session, page, application, pageContext, which are outside the scope of this tutorial.

#### 5.1  JSP comment <%-- comments --%>

JSP comments <%-- comments --%> are ignored by the JSP engine. For example,

<%-- anything but a closing tag here will be ignored -->

Note that HTML comment is <!-- comments -->. JSP expression within the HTML comment will be evaluated. For example,

<!-- HTML comments here <%= Math.random() %> more comments -->

#### 5.2  JSP Expression <%= Java Expression %>

JSP Expression can be used to insert a single Java expression directly into the response message. This expression will be placed inside a out.print() method. Hence, the expression will be evaluated and printed out as part of the response message.  Any valid Java expression can be used.  There is no semi-colon at the end of the expression. For examples:

<p>The square root of 5 is <%= Math.sqrt(5) %></p>

<h5><%= item[10] %></h5>

<p>Current time is: <%= new java.util.Date() %></p>

The above JSP expressions will be converted to:

out.write("<p>The square root of 5 is ");

out.print( Math.sqrt(5) );

out.write("</p>");

out.write("<h5>");

out.print( item[10] );

out.write("</h5>");

out.write("<p>Current time is: ");

out.print( new java.util.Date() );

out.write("</p>");

You can use the pre-defined variables, such as request, in the expressions. For examples:

<p>You have choose author <%= request.getParameter("author") %></p>

<%= request.getRequestURI() %>

<%= request.getHeader("Host") %>

#### 5.3  JSP Scriptlet <% Java statement(s) %>

JSP scriptlets allow you to do more complex operations than inserting a single Java expression (with the JSP expression). JSP scriptlets let you insert an arbitrary sequence of valid Java statement(s) into the service() method of the converted servlet. All the Java statements in a scriptlet are to be terminated with a semi-colon. For example:

<%

String author = request.getParameter("author");

if (author != null && !author.equals(""))) {

%>

<p>You have choose author <%= author %></p>

<%

}

%>

In the converted servlet, the above will be inserted into the service() method as follows:

String author = request.getParameter("author");

if (author != null && !author.equals(""))) {

out.write("<p>You have choose author ");

out.print( author );

out.write("</p>");

}

Notice that the Java codes inside a scriptlet are inserted exactly as they are written, and used as the programming logic. The HTML codes are passed to an out.write() method and written out as part of the response message.

#### 5.4  JSP Directive <%@ page|include ... %>

JSP directives provide instructions to the JSP engine. The syntax of the JSP directive is:

<%@ directive\_name

attribute1="value1"

attribute2="value2"

......

attributeN="valueN" %>

##### JSP page Directive

The "page" directive lets you import classes and customize the page properties. For examples,

<%-- import package java.sql.\* -->

<%@ page import="java.sql.\*" %>

<%-- Set the output MIME type -->

<%@ page contentType="image/gif" %>

<%-- Set an information message for getServletInfo() method -->

<%@ page info="Hello-world example" %>

##### JSP include Directive

The "include" directive lets you include another file(s) at the time when the JSP page is compiled into a servlet. You can include any JSP files, or static HTML files. You can use include directive to include navigation bar, copyright statement, logo, etc. on every JSP pages. The syntax is:

<%@ include file="url" **%>**

For example:

<%@ include file="header.html" %>

......

<%@ include file="footer.html" %>

### 6.  JSP Database Example

Let's revisit our e-shop, which was created using Java Servlet.

##### Database

Database: **ebookshop**

Table: **books**

+-------+----------------------------+---------------+---------+-------+

| **id**  | **title** | **author** | **price** | **qty**  |

| (INT) | (VARCHAR(50)) | (VARCHAR(50)) | (FLOAT) | (INT) |

+-------+----------------------------+---------------+---------+-------+

| 1001 | Java for dummies | Tan Ah Teck | 11.11 | 11 |

| 1002 | More Java for dummies | Tan Ah Teck | 22.22 | 22 |

| 1003 | More Java for more dummies | Mohammad Ali | 33.33 | 33 |

| 1004 | A Cup of Java | Kumar | 44.44 | 44 |

| 1005 | A Teaspoon of Java | Kevin Jones | 55.55 | 55 |

+-------+----------------------------+---------------+---------+-------+

##### Querying - "query.jsp"

Let's create the query page called "query.jsp".

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75 | <html>  <head>  <title>Book Query</title>  </head>  <body>  <h1>Another E-Bookstore</h1>  <h3>Choose Author(s):</h3>  <form method="get">  <input type="checkbox" name="author" value="Tan Ah Teck">Tan  <input type="checkbox" name="author" value="Mohd Ali">Ali  <input type="checkbox" name="author" value="Kumar">Kumar  <input type="submit" value="Query">  </form>    <%  String[] authors = request.getParameterValues("author");  if (authors != null) {  %>  <%@ page import = "java.sql.\*" %>  <%  Connection conn = DriverManager.getConnection(  "jdbc:mysql://localhost:8888/ebookshop", "myuser", "xxxx"); // <== Check!  // Connection conn =  // DriverManager.getConnection("jdbc:odbc:eshopODBC"); // Access  Statement stmt = conn.createStatement();    String sqlStr = "SELECT \* FROM books WHERE author IN (";  sqlStr += "'" + authors[0] + "'"; // First author  for (int i = 1; i < authors.length; ++i) {  sqlStr += ", '" + authors[i] + "'"; // Subsequent authors need a leading commas  }  sqlStr += ") AND qty > 0 ORDER BY author ASC, title ASC";    // for debugging  System.out.println("Query statement is " + sqlStr);  ResultSet rset = stmt.executeQuery(sqlStr);  %>  <hr>  <form method="get" action="order.jsp">  <table border=1 cellpadding=5>  <tr>  <th>Order</th>  <th>Author</th>  <th>Title</th>  <th>Price</th>  <th>Qty</th>  </tr>  <%  while (rset.next()) {  int id = rset.getInt("id");  %>  <tr>  <td><input type="checkbox" name="id" value="<%= id %>"></td>  <td><%= rset.getString("author") %></td>  <td><%= rset.getString("title") %></td>  <td>$<%= rset.getInt("price") %></td>  <td><%= rset.getInt("qty") %></td>  </tr>  <%  }  %>  </table>  <br>  <input type="submit" value="Order">  <input type="reset" value="Clear">  </form>  <a href="<%= request.getRequestURI() %>"><h3>Back</h3></a>  <%  rset.close();  stmt.close();  conn.close();  }  %>  </body>  </html> |

##### Explanations

1. This HTML page has a form with 3 checkboxes. The "name=value" pair of the checkboxes is "author=so\_and\_so". No "action" attribute is specified, hence, it defaulted to current page. The processing script is contained in the same page.
2. The method request.getParameter("author") is used to check if the query parameter "author" exists.  "author" is absent during the first reference of the page.
3. The <%@ page .. %> contains a JSP "page" directive to import the java.sql package.
4. The scriptlet performs the database query operation. The steps are:
   1. Establish a database connection via a java.sql.Connection object;
   2. Allocate a java.sql.Statement object under the Connection;
   3. Prepare a SQL SELECT string;
   4. Execute the SQL SELECT using executeQuery() method. The result of query is returned in an object of java.sql.ResultSet;
   5. Process the ResultSet row by row via ResultSet.next();
   6. Free resources and close the Connection.
5. The query result is tabulated in a HTML table. Note the mixing of HTML and Java in producing the table.

Notice that JSP carries out the presentation much better and neater than servlet. The presentation can be changed easily with JSP. The JSP pages can be created and modified using a WYSIWYG web authoring tool and reload to see the effect on the presentation.  Whereas, in the case of servlet, you have to explicitly code all the HTML tags inside the servlet program and re-compile the program.

##### Ordering - "order.jsp"

Let's write the "order.jsp" for processing the order, by updating the appropriate records in the database.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | <html>  <head>  <title>Order Book</title>  </head>    <body>  <h1>Another E-Bookstore</h1>  <h2>Thank you for ordering...</h2>    <%  String[] ids = request.getParameterValues("id");  if (ids != null) {  %>  <%@ page import = "java.sql.\*" %>  <%  Connection conn = DriverManager.getConnection(  "jdbc:mysql://localhost:8888/ebookshop", "myuser", "xxxx"); // <== Check!  // Connection conn =  // DriverManager.getConnection("jdbc:odbc:eshopODBC"); // Access  Statement stmt = conn.createStatement();  String sqlStr;  int recordUpdated;  ResultSet rset;  %>  <table border=1 cellpadding=3 cellspacing=0>  <tr>  <th>Author</th>  <th>Title</th>  <th>Price</th>  <th>Qty In Stock</th>  </tr>  <%  for (int i = 0; i < ids.length; ++i) {  // Subtract the QtyAvailable by one  sqlStr = "UPDATE books SET qty = qty - 1 WHERE id = " + ids[i];  recordUpdated = stmt.executeUpdate(sqlStr);  // carry out a query to confirm  sqlStr = "SELECT \* FROM books WHERE id =" + ids[i];  rset = stmt.executeQuery(sqlStr);  while (rset.next()) {  %>  <tr>  <td><%= rset.getString("author") %></td>  <td><%= rset.getString("title") %></td>  <td>$<%= rset.getInt("price") %></td>  <td><%= rset.getInt("qty") %></td>  </tr>  <% }  rset.close();  }  stmt.close();  conn.close();  }  %>  </table>  <a href="query.jsp"><h3>BACK</h3></a>  </body>  </html> |

# JSP Program Examples: Registration & Login Form

In this tutorial, we are going develop sample programs with JSP and using MVC architecture.

Following Program Examples, will be developed -

* [Registration form](https://www.guru99.com/jsp-example.html#1)
* [Login and Logout form](https://www.guru99.com/jsp-example.html#2)

## Using registration form through JSP

In Registration form, we will have a form to fill all the details which will contain name, username, password, address, contact number, etc.

This form will help us to register with the application. They take all our details and store it in a database or cache.

In this example, we are going to take "Guru Registration form" which has the following fields:

1. First Name
2. Last Name
3. Username
4. Password
5. Address
6. Contact Number

After filling all these details we have submit button, on click of that button all the details will be stored.

Register\_1.jsp

1. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2. pageEncoding="ISO-8859-1"%>
3. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7. <title>Guru Registration Form</title>
8. </head>
9. <body>
10. <h1>Guru Register Form</h1>
11. <form action="guru\_register" method="post">
12. <table style="with: 50%">
13. <tr>
14. <td>First Name</td>
15. <td><input type="text" name="first\_name" /></td>
16. </tr>
17. <tr>
18. <td>Last Name</td>
19. <td><input type="text" name="last\_name" /></td>
20. </tr>
21. <tr>
22. <td>UserName</td>
23. <td><input type="text" name="username" /></td>
24. </tr>
25. <tr>
26. <td>Password</td>
27. <td><input type="password" name="password" /></td>
28. </tr>
29. <tr>
30. <td>Address</td>
31. <td><input type="text" name="address" /></td>
32. </tr>
33. <tr>
34. <td>Contact No</td>
35. <td><input type="text" name="contact" /></td>
36. </tr></table>
37. <input type="submit" value="Submit" /></form>
38. </body>
39. </html>

**Explanation of the code:**

**Code Line 11:** Here we are taking a form name which has action i.e. the servlet to which the request will be processed and servlet name is guru\_register.java. The request will be processed through POST method.

**Code Line 14-16:** Here we are taking input type as text and name is first name

**Code Line 18-20:** Here we are taking input type as text and name is last name

**Code Line 22-24:** Here we are taking input type as text and name is username

**Code Line 26-28:** Here we are taking input type as password(this will hide the password when typed) and name as password

**Code Line 30-32:** Here we are taking input type as text and name as address

**Code Line 34-36:** Here we are taking input type as text and name as contact

**Code Line 37:** Here we are taking a button of type submit and value is also submit. On click of this button the action will go to corresponding guru\_register servlet where all the parameter values will be passed in the request.

Guru\_register.java

1. package demotest;
3. import java.io.IOException;
5. import javax.servlet.RequestDispatcher;
6. import javax.servlet.ServletException;
7. import javax.servlet.http.HttpServlet;
8. import javax.servlet.http.HttpServletRequest;
9. import javax.servlet.http.HttpServletResponse;
11. /\*\*
12. \* Servlet implementation class guru\_register
13. \*/
14. public class guru\_register extends HttpServlet {
15. private static final long serialVersionUID = 1L;

18. protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19. // TODO Auto-generated method stub
20. String first\_name = request.getParameter("first\_name");
21. String last\_name = request.getParameter("last\_name");
22. String username = request.getParameter("username");
23. String password = request.getParameter("password");
24. String address = request.getParameter("address");
25. String contact = request.getParameter("contact");
27. if(first\_name.isEmpty() || last\_name.isEmpty() || username.isEmpty() ||
28. password.isEmpty() || address.isEmpty() || contact.isEmpty())
29. {
30. RequestDispatcher req = request.getRequestDispatcher("register\_1.jsp");
31. req.include(request, response);
32. }
33. else
34. {
35. RequestDispatcher req = request.getRequestDispatcher("register\_2.jsp");
36. req.forward(request, response);
37. }
38. }
40. }

**Explanation of the code:**

**Code Line 14:** Here we defining guru\_servlet which is extending HttpServlet.

**Code Line 18:** This action doPost() method which will be called when we mention POST in action attribute in the above JSP form.

**Code Line 20-25:**Here we are fetching the values from request i.efirst\_name, last\_name , username, password, address and contact using request.getParameter.

**Code Line 27-32:** Here we are taking if condition where we check any of the parameters which are fetched from request as whether they are empty or not. If any of the parameter is empty then it will enter this condition ( first\_name.isEmpty() || last\_name.isEmpty || username.isEmpty || password.isEmpty || address.isEmpty || contact.isEmpty()) and we have to fetch RequestDispatcher object using request object which will forward request to register\_1.jsp. Here we are also including request and response objects.

**Code Line 33-37:** This case will execute when any of the parameter is not empty .We will have to fetch requestDispatcher object using request object which will forward request to register\_2.jsp.Here we are forwarding request and response objects.

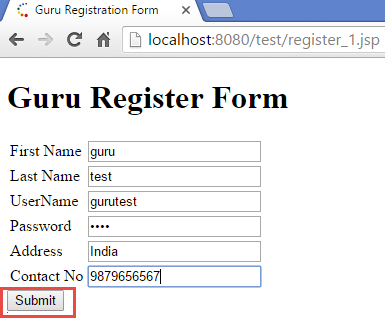
Register\_2.jsp

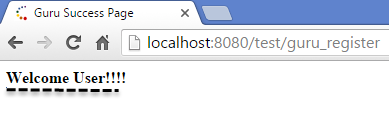
1. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2. pageEncoding="ISO-8859-1"%>
3. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7. <title>Guru Success Page</title>
8. </head>
9. <body>
10. <a><b>Welcome User!!!!</b></a>
11. </body>
12. </html>

**Explanation of the code:**

**Code Line 10:** Here we are saying welcome user. This JSP will be called when all the parameters are filled.

When you execute the above code , you get the following output:

[](https://www.guru99.com/images/jsp/022916_0744_JSPExample4.png)

[](https://www.guru99.com/images/jsp/022916_0744_JSPExample5.png)

**Output:**

When we click on register\_1.jsp, we will get a form which will have details like first name, last name, username, password, address, contact. All the details have been filled. When we click on submit button then we get the message as "Welcome User"

## Login and logout form

Like registration form we will have a login and logout form.

In this example, we have taken Login form where we have two fields "username" and "password" with a submit button.

When we click on submit button then we get welcome message with a logout button.

When we click on logout button then we get back to login form.

Register\_3.jsp

1. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2. pageEncoding="ISO-8859-1"%>
3. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7. <title>Guru Login Form</title>
8. </head>
9. <body>
10. <form action="guru\_login" method="post">
11. <table style="with: 50%">
13. <tr>
14. <td>UserName</td>
15. <td><input type="text" name="username" /></td>
16. </tr>
17. <tr>
18. <td>Password</td>
19. <td><input type="password" name="password" /></td>
20. </tr>
21. </table>
22. <input type="submit" value="Login" /></form>
23. </body>
24. </html>

**Explanation of the code:**

**Code Line 10:**Here we are taking a form name which has action i.e. servlet to which it has passed is guru\_login.java. The method through which it will pass its POST.

**Code Line 13-16:** Here we are taking an input field "username" which is of the type text.

**Code Line 17-20:** Here we are taking an input field "password" which is of the type password.

**Code Line 22:** Here we are taking a "submit" button with the value"Login" on which we click then it goes to servlet guru\_login where both the fields are taken using request object.

Guru\_login.java(servlet)

1. package demotest;
3. import java.io.IOException;
5. import javax.servlet.RequestDispatcher;
6. import javax.servlet.ServletException;
7. import javax.servlet.http.HttpServlet;
8. import javax.servlet.http.HttpServletRequest;
9. import javax.servlet.http.HttpServletResponse;
11. /\*\*
12. \* Servlet implementation class guru\_login
13. \*/
14. public class guru\_login extends HttpServlet {
16. public guru\_login() {
17. super();
18. // TODO Auto-generated constructor stub
19. }
21. protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
22. // TODO Auto-generated method stub
23. String username = request.getParameter("username");
24. String password = request.getParameter("password");
25. if(username.isEmpty() || password.isEmpty() )
26. {
27. RequestDispatcher req = request.getRequestDispatcher("register\_3.jsp");
28. req.include(request, response);
29. }
30. else
31. {
32. RequestDispatcher req = request.getRequestDispatcher("register\_4.jsp");
33. req.forward(request, response);
34. }
35. }
37. }

**Explanation of the code:**

**Code Line 5-9:** Here we are importing necessary imports in the code.

**Code Line 14:** Here we are taking guru\_login servlet which extends HttpServlet.

**Code Line 21:** Here we are using doPost() method as in the form we are using POST method.

**Code Line 23-24:** Here we taking parameters using request object i.e. username and password.

**Code Line 25-29:** In this way, we are taking "if" condition where we are checking username and password whether they are empty or not.In this case if it is empty then we are getting requestdispatcher object which forwards to register\_3.jsp with request and response objects.

**Code Line 30-34:** This will be executed if both are not empty then it forwards the request to register\_4.jsp with request and response objects.

Register\_4.jsp >

1. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2. pageEncoding="ISO-8859-1"%>
3. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7. <title>Guru Logged In</title>
8. </head>
9. <body>
10. <table style="with: 50%">
11. <tr><td>
12. <% String username = request.getParameter("username"); %>
13. <a>Welcome <% out.println(username); %> User!!!! You have logged in.</a></td></tr>
14. <tr></tr><tr><td></td><td></td><td><a href="register\_3.jsp"><b>Logout</b></a></td></tr>
15. </table>
16. </body>
17. </html>

**Explanation of the code:**

**Code Line 12:** Here we are getting parameter "username" from the request object in the string object username.

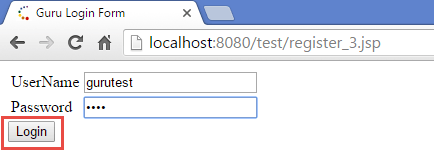
**Code Line 13:** Here we have a welcome message with the username.

**Code Line 14:** Here we link to logout the form which redirects to register\_3.jsp.

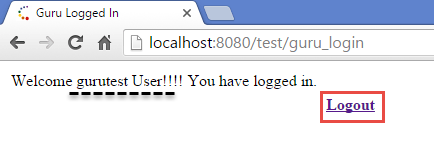
When you execute the above code then you get the following output:

**Output:**

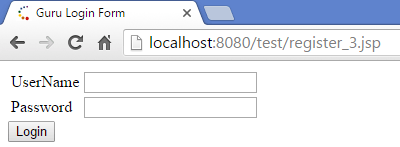
Here when we click on register\_3.jsp we get two fields"username" and "password" with a login button.

[](https://www.guru99.com/images/jsp/022916_0744_JSPExample9.png)

After clicking on the Login button you get the below message with a button of Logout.

[](https://www.guru99.com/images/jsp/022916_0744_JSPExample10.png)

When you click on logout button you go back to login page

[](https://www.guru99.com/images/jsp/022916_0744_JSPExample11.png)