

## Assignment - 6

Title :- Add dynamic web application essence using PHP, HTML and MySQL.

Objective :- To understand the principle and methodologies of PHP web based applications development process.

Problem Statement:- Design and develop dynamic web application using PHP and MySQL as a backend for employee data with insert, delete, view and update operations.

Software requirement :-

1. Ubuntu 64 bit / windows XP,
2. XAMPP Server.

Hardware Requirement:-

Intel p4 Machine with 1GB RAM & 32GB HD.

Theory :-

1. PHP

The PHP Hypertext Preprocessor began as a little open source venture that advanced as an ever increasing number of individuals discovered how valuable it was.

PHP is a server side Scripting dialect that is installed in HTML. It is utilized to oversee dynamic

Substance, databases, Session following even from whole internet business locales. It is incorporated with various prevalent databases, including MySQL, PostgreSQL Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP performs framework capacities i.e. from documents on a framework it can be make, open, read, compare, and close them. PHP can deal with frames i.e. accumulate information from records, Splice information to a document, through email to get can send information, return information to the client.

Example:

"Hello world" Script in PHP

To get a feel for PHP, first start with Simple PHP Scripts.

As mentioned earlier PHP is embedded in HTML. That means that in amongst your normal HTML you'll have PHP statements like this

```
<html>
  <head>
    <title>Hello world</title>
```

```

</head>
<body>
<?php echo ("Hello PHP"); ?>
</body>
</html>

```

## 2. MySQL :-

MySQL is the most famous Open Source Relational SQL Database Management System. MySQL is outstanding amongst other RDBMSs being utilized for creating different online programming applications.

## 3 What is a Database?

A database is different application that stores a gathering of information. Every database has at least one unmistakable APIs for making, getting to, overseeing, seeking and reworking the information it holds.

These days, we utilize Social database administration frameworks (RDBMS) to store and oversee tremendous volume of information. This is called social database since every one of the information is put away into various tables and relations are set up utilizing essential keys or different keys known as foreign keys.

A Relational Database Management System (RDBMS) is a product that :

- Empowers you to execute a database with tables, segments & records.
- Ensures the Referential Integrity between columns of different tables.
- Updates the lists naturally.
- Deciphers a SQL inquiry & consolidates data from different tables.

MySQL Database :-

MySQL is a quick, simple-to-utilize RDBMS being utilized for some little and huge organizations. MySQL is produced, showcased and upheld by MySQL AB, which is a Swedish organization. MySQL is winding up so famous as a result of numerous great reasons.

- MySQL is discharged under an open-source permit. So you don't have anything to pay to utilize it.
- MySQL is a capable program in its own particular right. It handles a huge subset of the usefulness of the most costly and intense database bundles.

- MySQL utilizes a standard type of the outstanding SQL information dialect.
- MySQL works rapidly and functions admirably even with extensive informational indexes.
- MySQL is adaptable. The open source GPL permits enables developers to alter the MySQL programming to fit their own particular pr surroundings.

### Conclusion:-

In this assignment, we have studied how to design and develop small web application using PHP Script, XAMPP Server with apache Server and MySQL as backend.

Design and develop a dynamic web application using php, and mysql as a backend for employee data with insert , update,delete,view operations.

```
CREATE TABLE IF NOT EXISTS `new_record` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `trn_date` datetime NOT NULL,
  `empname` varchar(50) NOT NULL,
  `salary` int(11) NOT NULL,
  `submittedby` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
);
```

```
<?php
require('db.php'); include("auth.php"); ?>

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Dashboard - Secured Psalary</title>
<link rel="stylesheet" href="css/style.css" />
</head>
<body>
<div class="form">
<p>Welcome to Dashboard.</p>
<p><a href="index.php">Home</a><p>
<p><a href="insert.php">Insert New Record</a></p>
<p><a href="view.php">View Records</a><p>
<p><a href="logout.php">Logout</a></p>
</div>
</body>
</html>
<?php
require('db.php'); include("auth.php"); $status = "";
```

```
if(isset($_POST['new']) && $_POST['new']==1){

$trn_date = date("Y-m-d H:i:s");

$empname =$_REQUEST['empname'];

$salary = $_REQUEST['salary'];

$submittedby = $_SESSION["userempname"];

$ins_query="insert into new_record

(`trn_date`,`empname`,`salary`,`submittedby`)values

('$trn_date','$empname','$salary','$submittedby')"; mysqli_query($con,$ins_query)

or die(mysql_error());

$status = "New Record Inserted Successfully.

</br></br><a href='view.php'>View Inserted Record</a>"; }

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Insert New Record</title>

<link rel="stylesheet" href="css/style.css" />

</head>

<body>

<div class="form">

<p><a href="dashboard.php">Dashboard</a>

| <a href="view.php">View Records</a>

| <a href="logout.php">Logout</a></p>

<div>

<h1>Insert New Record</h1>

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

<input type="hidden" empname="new" value="1" />

<p><input type="text" empname="empname" placeholder="Enter Empname" required /></p>

<p><input type="text" empname="salary" placeholder="Enter Salary" required /></p>

<p><input empname="submit" type="submit" value="Submit" /></p>

</form>

<p style="color:#FF0000;"><?php echo $status; ?></p>

</div>

</div>

</body>

</html>
```

# Insert New Record

```
<?php
require('db.php'); include("auth.php"); ?>

<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<title>View Records</title>
<link rel="stylesheet" href="css/style.css" />
</head>
<body>
<div class="form">
<p><a href="index.php">Home</a>
| <a href="insert.php">Insert New Record</a>
| <a href="logout.php">Logout</a></p>
<h2>View Records</h2>
<table width="100%" border="1" style="border-collapse:collapse;">
<thead>
<tr>
<th><strong>Empid</strong></th>
<th><strong>Empname</strong></th>
<th><strong>Salary</strong></th>
<th><strong>Edit</strong></th>
<th><strong>Delete</strong></th>
</tr>
</thead>
```

```

<tbody>
<?php
$count=1;

$sel_query="Select * from new_record ORDER BY id desc;"

$result      =      mysqli_query($con,$sel_query);      while($row      =
mysqli_fetch_assoc($result)) { ?> <tr><td align="center"><?php echo $count;
?></td>
<td align="center"><?php echo $row["empname"]; ?></td>
<td align="center"><?php echo $row["salary"]; ?></td>
<td align="center">
<a href="edit.php?id=<?php echo $row["id"]; ?>">Edit</a>
</td>
<td align="center">
<a href="delete.php?id=<?php echo $row["id"]; ?>">Delete</a>
</td>
</tr>
<?php $count++; } ?>
</tbody>
</table>
</div>
</body>
</html>

```

[Home](#) | [Insert New Record](#) | [Logout](#)

## View Records

S.No	Name	Age	Edit	Delete
1	Adnan	30	<a href="#">Edit</a>	<a href="#">Delete</a>
2	Imran	18	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Kashif	25	<a href="#">Edit</a>	<a href="#">Delete</a>
4	Kamran	25	<a href="#">Edit</a>	<a href="#">Delete</a>
5	Javed	23	<a href="#">Edit</a>	<a href="#">Delete</a>

<?php

```

require('db.php'); include("auth.php");

$id=$_REQUEST['id'];

$query = "SELECT * from new_record where id='".$id."'";
$result = mysqli_query($con, $query) or die ( mysqli_error());
$row = mysqli_fetch_assoc($result);

?>
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<title>Update Record</title>
<link rel="stylesheet" href="css/style.css" />
</head>
<body>
<div class="form">
<p><a href="dashboard.php">Dashboard</a>
| <a href="insert.php">Insert New Record</a>
| <a href="logout.php">Logout</a></p>
<h1>Update Record</h1>
<?php
$status = "";
if(isset($_POST['new']) && $_POST['new']==1)
{
$id=$_REQUEST['id'];
$trn_date = date("Y-m-d H:i:s");
$empname =$_REQUEST['empname'];
$salary =$_REQUEST['salary'];
$submittedby = $_SESSION["userempname"]; $update="update new_record set
trn_date='".$trn_date."',      empname='".$empname."',      salary='".$salary."',
submittedby='".$submittedby."' where id='".$id."'"; mysqli_query($con, $update) or
die(mysqli_error()); $status = "Record Updated Successfully. <br><br> <a
href='view.php'>View Updated Record</a>"; echo '<p
style="color:#FF0000;">'.$status.'</p>';
}else {
?>
<div>
<form empname="form" method="post" action="">
<input type="hidden" empname="new" value="1" />
<input empname="id" type="hidden" value="<?php echo $row['id'];?>" /> <p><input type="text"
empname="empname" placeholder="Enter Empname" required value="<?php echo
$row['empname'];?>" /></p> <p><input type="text" empname="salary" placeholder="Enter Salary"
required value="<?php echo $row['salary'];?>" /></p>

```

```
<p><input empname="submit" type="submit" value="Update" /></p>
</form>
<?php } ?>
</div>
</div>
</body>
</html>
```

# Update Record

Adnan

30

Update

```
<?php
require('db.php');

$id=$_REQUEST['id'];

$query = "DELETE FROM new_record WHERE id=$id"; $result =
mysqli_query($con,$query) or die ( mysqli_error()); header("Location: view.php");
?>
```

# Assignment - 7

Name - Bhushan Nikumbhe  
ROLL NO. 22  
Div - B

Page No.	1
Date	

Title :- Add dynamic web application essence using PHP, AJAX and MySQL

Objectives:- To understand the principles and methodologies of web based applications development process.

Problem statement:- Design and develop dynamic web application using PHP, AJAX and MySQL as a back-end for employee data with insert and view operations.

Software Requirement :-

1. Ubuntu 64 bit / windows XP
2. XAMPP Server

Hardware Requirement :-

Intel P4 Machine with 1GB RAM & 32GB HDD.

Theory :-

AJAX remains for Asynchronous Javascript and XML. AJAX is another procedure for making better, speedier, and more intelligent dynamic web applications with the assistance of XML, HTML, CSS, and Javascript.

AJAX utilizes XHTML for content.

css for introduction, alongside Document Object Model and Javascript for dynamic substance to show.

Customary web applications transmit data to and from server utilizing Synchronous solicitations. It implies you round you round out a frame, hit Submit & get coordinated to another page with new data from the server. With AJAX, when you hit Submit, Javascript will influence a demand to the server, to decipher the outcomes and refresh the present screen. In the purest sense, the client could never realize that anything was even transmitted to the server.

AJAX instructional exercise cases ideas and cases of AJAX innovation for apprentices and experts.

AJAX is an acronym for Asynchronous Javascript and XML. It is a gathering of between related innovations like Javascript, DOM, XML, HTML, CSS and so forth.

AJAX enables you to send and get information nonconcurrently without reloading the page so it is quick.

AJAX speaks with Server utilizing XMLHttpRequest question. How about we

endeavor to comprehend the stream of AJAX or how AJAX functions by the picture shown beneath.

AJAX communicates with the server using XML HTTP Request Object. Let's try to understand the flow of AJAX or how ajax works by the image.

Technology / Tool :-  
AJAX, PHP and MySQL.

Conclusion :-

In this assignment, we have studied how to design and develop small web application using PHP java script, ajax, XAMPP server with apache server and MySQL as backend.

Assignment 7 : Design and develop dynamic web application using php and Ajax and MySQL as a back-end for employee data with insert and view operations.

**Database**

```
CREATE TABLE `tbl_employee` ( `id` int(11) NOT NULL,  
`name` varchar(50) NOT NULL,  
`address` text NOT NULL,  
`gender` varchar(10) NOT NULL,  
`designation` varchar(100) NOT NULL,  
`age` int(11) NOT NULL,  
`images` varchar(150) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
INSERT INTO `tbl_employee`(`name`, `age`, `city`) VALUES  
(‘Krushna Solapur’,20,’Pune’),  
(‘Krushna Belerao’,35,’Pune’),  
(‘Krushna Belerao’,35,’Pune’),  
(‘Krushna Belerao’,35,’Pune’),  
(‘Krushna Belerao’,35,’Pune’),  
(‘Krushna Belerao’,12,’vvvv’),  
(‘Krushna ’,0,’ ’),  
(‘Krushna ’,25,’solapur’),
```

- Indexes for dumped tables

- Indexes for table `tbl\_employee`

-  
ALTER TABLE `tbl\_employee` ADD PRIMARY KEY (`id`);

```
- AUTO_INCREMENT for dumped tables

-
- AUTO_INCREMENT for table `tbl_employee`


--



ALTER TABLE `tbl_employee`


MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=206;
```

Database\_Connection.Php

```
<?php
```

```
//database_connection.php
```

```
$username = 'root';
$password = "";
$connect = new PDO( 'mysql:host=localhost;dbname=testing', $username, $password );
```

```
?>
```

```
<html>
<head>

<title>Insert or Add Data using jQuery Dialogify with PHP Ajax</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
<script src="https://cdn.datatables.net/1.10.12/js/jquery.dataTables.min.js"></script> <script
src="https://cdn.datatables.net/1.10.12/js/dataTables.bootstrap.min.js"></script> <link rel="stylesheet"
href="https://cdn.datatables.net/1.10.12/css/dataTables.bootstrap.min.css" />
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
<script src="https://www.jqueryscript.net/demo/Dialog-Modal-
Dialogify/dist/dialogify.min.js"></script>

</head>
<body>

<div class="container">
<br />
<h3 align="center">Insert or Add Data using jQuery Dialogify with PHP
Ajax</h3> <br />
<div class="panel panel-default">
```

```
<div class="panel-heading">
<div class="row">
<div class="col-md-6">
<h3 class="panel-title">Employee Data</h3>
</div>
<div class="col-md-6" align="right">
<button type="button" name="add_data" id="add_data" class="btn btn-success btnxs">Add</button>
</div>
</div>
</div>

<div class="panel-body">
<div class="table-responsive">
<span id="form_response"></span>
<table id="user_data" class="table table-bordered table-striped">
<thead>
<tr>
<td>Name</td>
<td>Age</td>
<td>City</td>
</tr>
</thead>
</table>
</div>
</div>
</div>
</div>
</body>
</html>
```

```
<script type="text/javascript" language="javascript"
>$(document).ready(function(){

var dataTable = $('#user_data').DataTable({
"processing":true,
"serverSide":true,
"order":[],
```

```
"ajax":{ url:"fetch.php",
  type:"POST"
},
"columnDefs":[
{
  "targets":[4],
  "orderable":false,
},
],
});

$(document).on('click', '.view', function(){
var id = $(this).attr('id'); var options = {
  ajaxPrefix: '', ajaxData: {id:id},
  ajaxComplete:function(){ this.buttons([
    {
      type: Dialogify.BUTTON_PRIMARY
    }]);
  }
};

new Dialogify('fetch_single.php', options)
.title('View Employee Details')
.showModal();
});

$('#add_data').click(function(){ var options = { ajaxPrefix:''
};

new Dialogify('add_data_form.php', options)
.title('Add New Employee Data')
.buttons([
{
  text:'Cancel', click:function(e){ this.close();
}
}, { text:'Insert',
  type:Dialogify.BUTTON_PRIMARY,
  click:function(e)
{
  var image_data = $('#images').prop("files")[0];

```

```

var form_data = new FormData();
form_data.append('images', image_data);
form_data.append('name', $('#name').val());
form_data.append('age', $('#age').val());
form_data.append('City', $('#City').val()); $.ajax({
method:"POST",
url:'insert_data.php',
data:form_data, dataType:'json', contentType:false,
cache:false, processData:false, success:function(data)
{ if(data.error != "")
{
$('#form_response').html('<div class="alert alert-danger">'+data.error+'</div>');
}
else
{
$('#form_response').html('<div class="alert alert-success">'+data.success+'</div>'); dataTable.ajax.reload();
}
}
});
});
```

]);showModal();});});

</script>

Fetch.PhP

```

<?php
include('database_connection.php');
$query = "";
$output = array();
$query .= "SELECT * FROM tbl_employee "; if(isset($_POST["search"]["value"]))
{
$query .= 'WHERE name LIKE "%'.$_POST["search"]["value"].'%" OR address LIKE
"%. $_POST["search"]["value"].%" OR gender LIKE "%'.$_POST["search"]["value"].'%" OR designation LIKE
"%. $_POST["search"]["value"].%" OR age LIKE "%'. $_POST["search"]["value"].'%" ';
```

```
}

if(isset($_POST["order"]))
{
    $query .= 'ORDER BY '.$_POST['order'][0]['column'].' '.$_POST['order'][0]['dir'].';

}
else
{
    $query .= 'ORDER BY id DESC';

}
if($_POST["length"] != -1)
{
    $query .= 'LIMIT ' . $_POST['start'] . ',' . $_POST['length'];
}

$statement = $connect->prepare($query);
$statement->execute();
$result = $statement->fetchAll();
$data = array();
$filtered_rows = $statement->rowCount(); foreach($result as $row)
{
    $sub_array = array();
    $sub_array[] = $row["name"];
    $sub_array[] = $row["gender"];
    $sub_array[] = $row["designation"];
    $sub_array[] = $row["age"];
    $sub_array[] = '<button type="button" name="view" id="'.$row["id"].'" class="btn btnprimary btn-xs view">View</button>';
    $data[] = $sub_array;
}

function get_total_all_records($connect)
{
    $statement = $connect->prepare("SELECT * FROM tbl_employee"); $statement->execute(); $result = $statement->fetchAll(); return $statement->rowCount();
}

$output = array(
    "draw"    => intval($_POST["draw"]),
    "recordsTotal" => $filtered_rows,
```

```
"recordsFiltered" => get_total_all_records($connect),
"data"      => $data
);
echo json_encode($output);
?>
```

## Fetch\_Single.Php

```
<?php

//fetch_single.php

include('database_connection.php');

if(isset($_GET["id"]))
{
    $query = "SELECT * FROM tbl_employee WHERE id = '".$_GET["id"]."'";

    $statement = $connect->prepare($query);
    $statement->execute();
    $result = $statement->fetchAll(); $output = '<div class="row">';
    foreach($result as $row)
    {
        $images = ""; if($row["images"] != "") {
            $images = '';
        }
        else
        {
            $images = '';
        }
        $output .= '
<div class="col-md-3">
<br />
'.$images.'
```

```
</div>

<div class="col-md-9">
<br />

<p><label>Name :&nbsp;</label>' . $row["name"] . '</p>
<p><label>Address :&nbsp;</label>' . $row["address"] . '</p>
<p><label>Gender :&nbsp;</label>' . $row["gender"] . '</p>
<p><label>Designation :&nbsp;</label>' . $row["designation"] . '</p>
<p><label>Age :&nbsp;</label>' . $row["age"] . ' years</p>
</div>
</div><br />
';

}

echo $output;
}

?>
```

Add\_Data\_Form.Php

```
<div class="form-group">
<label>Enter Employee Name</label>
<input type="text" name="name" id="name" class="form-control" />
</div>

<div class="form-group">
<label>Enter Employee Address</label>
<textarea name="address" id="address" class="form-control"></textarea>
</div>

<div class="form-group">
<label>Enter Employee Gender</label>
<select name="gender" id="gender" class="form-control">
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
</div>

<div class="form-group">
<label>Enter Employee Desingation</label>
<input type="text" name="designation" id="designation" class="form-control" />
</div>
```

```
<div class="form-group">
<label>Enter Employee Age</label>
<input type="text" name="age" id="age" class="form-control" /
></div>

<div class="form-group">
<label>Select Employee Image</label>
<input type="file" name="images" id="images" />
</div>
```

insert\_data.php

```
<?php
```

```
//insert_data.php
```

```
include('database_connection.php');
```

```
if(isset($_POST["name"]))
{
    $error = "";
    $success = "";
    $name = "";
    $address = "";
    $designation = "";
    $age = "";
    $images = "";
    $gender = $_POST["gender"];
    if(empty($_POST["name"]))
    {
        $error .= '<p>Name is Required</p>';
    }
    else
    {
        $name = $_POST["name"];
    }
    if(empty($_POST["address"]))
    {
```

```
$error .= '<p>Address is Required</p>';
}

else
{
    $address = $_POST["address"];
}

if(empty($_POST["designation"]))
{
    $error .= '<p>Designation is Required</p>';
}

else
{
    $designation = $_POST["designation"];
}

if(empty($_POST["age"]))
{
    $error .= '<p>Age is Required</p>';
}

else
{
    $age = $_POST["age"];
}

if(isset($_FILES["images"]["name"]) && $_FILES["images"]["name"] != "")
{
    $image_name = $_FILES["images"]["name"];
    $array = explode(".", $image_name);
    $extension = end($array);

    $temporary_name = $_FILES["images"]["tmp_name"];
    $allowed_extension = array("jpg", "png"); if(! in_array($extension, $allowed_extension)) {

        $error .= '<p>Invalid Image</p>';
    }

    else
    {
        $images = rand() . '.' . $extension;
        move_uploaded_file($temporary_name, 'images/' . $images);
    }
}
```

```
} } if($error == "")  
{  
$data = array(  
    ':name' => $name,  
    ':address' => $address,  
    ':gender' => $gender,  
    ':designation' => $designation,  
    ':age' => $age,  
    ':images' => $images  
);  
  
$query = "  
INSERT INTO tbl_employee  
(name, address, gender, designation, age, images)  
VALUES (:name, :address, :gender, :designation, :age, :images) ";  
$statement = $connect->prepare($query);  
$statement->execute($data);  
$success = 'Employee Data Inserted';  
}  
  
$output = array(  
    'success' => $success,  
    'error' => $error  
);  
echo json_encode($output);  
}  
  
?>
```

OUTPUT

Add New Data

Name	Age	City
krushna Solapur	20	pune
krushna Belerao	35	Pune
krushna	12	vvvv
krushna	0	
krushna	25	solapur

## Assignment - 8

Title :- Design and develop any web application using struct framework.

## Objectives:-

1. To impact the efficient and available client side and server side technologies.
2. To implement the communication between computing nodes using client side and server side technologies.
3. To design and implement the web services with content management.

## Problem Statement:-

Create a login module for the web application using struct framework.

## Software &amp; Hardware Requirements:-

Software's : Java 7 or higher,  
Apache tomcat 7 or higher, struct API's , Eclipse IDE.

## Theory :-

The frameworks plays a vital role in industries for example manageable and well designed application development as well as enterprise application development.

The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, Resource Bundles, and XML, as well as various Jakarta

The Struts system gives the undetectable underpinnings each expert web applications needs to survive. Strut causes you make an extensible advancement condition for your application, in view of distributed guidelines and demonstrated outlines designs.

#### - The Model-View-Controller Architecture

The "Modelview-Controller" is a way to build applications that promotes complete separation between business logic and presentation. It is not specific to web applications or Java, or J2EE, but it can be applied to building J2EE web application.

The view is the user interface the screens that the end user of the application, actually see, and interact with. In a J2EE application views are JSP files for collecting user input, you will have a JSP that generates an HTML page that

contains one or more HTML form. For displaying output (like a report), you will have a JSP generates an HTML page that probably contains one or more HTML tables. Each of these is a view, a way for the end user to interact with the system, putting data in & getting data out.

What is Struts?

Struts is a framework that advances the utilization of the Model-view-controller engineering for planning substantial scale applications. The structure incorporates an arrangement of custom tag libraries & their related Java classes, alongside different utility classes. The most intense part of Struts is its help for making and preparing electronics structures, we will perceive how this functions later in this section.

Struts Tags

Almost all tags provided by Struts framework use the following attributes:

- **Attributes** : used for.
- **ID** : the name of a bean for temporary use by the tag.
- **name** : the name of a pre-existing bean for use with the tag.
- **property** : the property of the bean named in the name attribute for use with the tag.
- **scope** : the scope to search for the bean name in the name attribute.

### Conclusion :-

Hence we have successfully tested the Struts framework and tested the results.

Assignment 8 : Create a login module for the web application using struts framework.

### **LoginForm.java**

```
package com.example.javawebtutor.form; import javax.servlet.http.HttpServletRequest;  
  
import org.apache.struts.action.ActionForm; import  
org.apache.struts.action.ActionMapping; public class LoginForm extends ActionForm {  
  
    private String userName = null; private String password = null;  
  
    public String getUserName() {  
        return userName;  
    }  
  
    public void setUserName(String userName) {  
        this.userName = userName;  
    }  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String password) {  
        this.password = password;  
    }  
  
    @Override public void reset(ActionMapping mapping, HttpServletRequest request) {  
        this.password = null;  
    }  
}
```

### **LoginAction.java**

```
package com.example.javawebtutor.action;  
  
import javax.servlet.http.HttpServletRequest; import  
javax.servlet.http.HttpServletResponse;
```

```

import org.apache.struts.action.Action; import org.apache.struts.action.ActionForm; import
org.apache.struts.action.ActionForward; import org.apache.struts.action.ActionMapping; import
com.example.javawebtutor.form.LoginForm;

public class LoginAction extends Action

{ @Override public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception {

    LoginForm loginForm = (LoginForm) form;

        if (loginForm.getUserName() == null || loginForm.getPassword() == null
    |      !loginForm.getUserName().equalsIgnoreCase("Mukesh")
    |      !loginForm.getPassword().equals("kumar")) { return

mapping.findForward("success"); } else

    return mapping.findForward("failure");

}

}

```

## Struts-config.xml

```

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

><!DOCTYPE struts-config PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration
1.3//EN" "http://struts.apache.org/dtds/struts-config\_1\_3.dtd

```

## Login.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

```

```

http://www.w3.org/TR/html4/loose.dtd"> <html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Example</title>
</head>
<body>

<html:form action="/login" focus="userName">

Username : <html:text property="userName" />

<br>

Password : <html:password property="password" />

> <br>

<html:submit value="login" />

</html:form>

</body> </html>

```

## **Success.jsp**

```

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Successful Login Page</title>
</head>
<body>
<h2>Welcome Boss</h2>
</body>
</html>

```

## **Index.jsp**

```
<jsp:forward page="/login.jsp"></jsp:forward>
```

Login Example

localhost:9999/LoginExampleStruts1/

Apps technical live tv Outlook Web App Google Structured D...

Username :

Password :

Successful Login Page

localhost:9999/LoginExampleStruts1/success.jsp

Apps technical live tv Outlook Web App Google Structured D...

Do you want Google Chrome to save your password?

Welcome Boss

# Assignment - 9

Name - Bhushan Nikumbre

Roll No. 22

Div - B

Page No.	1
Date	

Title :- Design and develop any web application using Ja AngularJS.

Objectives:-

1. Understand the design of single-page application and how AngularJS facilitates their development.
2. Master AngularJS expressions, filters, and scopes.
3. Build Angular Forms.
4. Elegantly implement Ajax in your AngularJS applications.
5. Write AngularJS directive.

Problem Statement:-

Create an application for Bill Payment Record using AngularJS.

Software & Hardware Requirements:-

Eclipse IDE / Notepad / Notepad++,

Modern web browser.

Theory:-

AngularJS is an open source web application framework. It was initially created in 2009 by Misko Hevery and Adam Abrons. It is presently kept up by Google. It gives you a chance to utilize HTML as your Layout dialect and gives you a

chance to stretch out HTML's linguistic structure to express your application parts plainly and compactly

### General Features -

- AngularJS is a productive system that can make Rich Internet Applications (RIA).
- AngularJS gives designer a choice to compose customer side application utilizing JavaScript in a spacial Model View controller (MVC) way.
- Application written in Javascript are cross-platform agreeable. AngularJS consequently handles Javascript code reasonable for every program.
- By and large, AngularJS is a system to assemble expensive scale, elite and simple to keep up web application.

### Core Features

1. Data-authoritative :- It is the programmed synchronization of information amongst model and view parts.
2. Scope : These are objects that allude to the model. They go about as paste amongst controller and view.
3. Controller :- These are Javascript capabilities bound to specific degree

4. Services :- AngularJS accompanies a few implicit administrations, for example \$http to make a XMLHttpRequest. These are singleton objects which are instantiated just once in application.
5. Filters :- These select a subset of things from a cluster and return another explicit.
6. Directives :- Directives are markers on DOM components.
7. Routing :- It is idea of exchanging sees.
8. Deep Learning Linking :- Deep linking permits to encode the condition of use in the URL with the goal that it can be bookmarked.

### Model View Controller

Model view controller or MVC as it is famously called, is a popular configuration design for creating web applications. A model view controller design is comprised of the accompanying three sections.

- Model - It is the most minimal level of the example in charge of looking after information.
- View - It is in charge of showing all or part of the information to the client.
- Controller - It is a product case that controls the connections between the Model and View.

MVC is mainstream since it includes application rational frame by UI layer and backing) detachment of concerns. The controller gets all solicitations for the application and afterward works with the model to set up any information required by the view.

AngularJS is a MVC based structure

- An AngularJS application binds the value(s) of AngularJS application data to HTML input control
- An AngularJS application comprise of following three essential parts -  
ng-app - This directive defines and links an AngularJS application to HTML.

- `ng-bind` - This directive binds the AngularJS Application data to HTML tags.

Conclusion :-

With the help of this assignment it is helpful to understand features of AngularJS, MVC mode structure and its use in advanced web programming is studied.

## Assignment 9 : Create an application for Bill Payment Record using AngularJs.

```

<!DOCTYPE html>
<html ng-app="billpayApp">
<head>
<title>AngularJS First Application : Responding to User
</title><link href="bootstrap.css" rel="stylesheet" />
<link href="bootstrap-theme.css" rel="stylesheet" />
><script src="angular.js"></script><script>

var dataModel = { customer: "Tintin", items:[{ bill:"Electricity",
status:false},
{ bill:"Internet(Wi/fi)", status:false }, { bill:"Parking Charges", status:false }, { bill:"Phone", status:true},
{ bill:"House Tax", status:true}]};

};

var billpayApp = angular.module("billpayApp", []);

billpayApp.controller("billPayCtrl", function($scope){ /* first argument is name of Controller,
second is a function to be called to define the functionality of controller*/
$scope.billpay = dataModel // property billpay on $scope service object, and assign model to it
$scope.dueBills = function() {
// dueBills is the behaviour
var counter = 0; angular.forEach($scope.billpay.items, function(item){
if(!item.status) {counter++} // checks if status is false, and then increases counter by one
});
return counter;
}

$scope.redFlag = function(){ return $scope.dueBills() < 2 ? "label-success" : "label-danger";
}

$scope.newBills = function(billName){
$scope.billpay.items.push({ bill: billName, status: false});
}

```

```

// Adds new items to the model
}

});

</script>

</head>

<body ng-controller="billPayCtrl">

<div class="page-header">
<h1>{{billpay.customer}}'s Bills to Be Paid -
<span class="label " ng-class="redFlag()" ng-hide="dueBills() == 0">
<!-- ng-hide hides element if the expression within is true -->

{{dueBills()}} <!-- Behaviour is called using Parentheses, it gets data from the scope -->

</span></h1>
</div>

<div class="panel">
<div class="input-group">
<input class="form-control" ng-model="billName"/>
<!-- ng-model is used to create the specified property -->

<span class="input-group-btn">
<button class="btn btn-danger" ng-click="newBills(billName)">+Add+</button>
<!-- the directive ng-click executes the expression when click event
is triggered -->
</span>
</div>
<table class="table table-striped">

<thead>
<tr>
<th>Bill Name</th>
<th>Status</th>
</tr>
</thead>

<tbody>

```

```

<tr ng-repeat="item in billpay.items">
  <td>{{item.bill}}</td>
  <td><input type="checkbox" ng-model="item.status"/></td>

</tr>
</tbody>

</table>
</div>

</body>
</html>

```

OUTPUT:

Add extra biller fields if any			
	Status	Status	Close
Electricity	<input type="checkbox"/>	false	
Internet(Wi-fi)	<input type="checkbox"/>	false	
Parking Charges	<input type="checkbox"/>	false	
Phone	<input checked="" type="checkbox"/>	true	
House Tax	<input checked="" type="checkbox"/>	true	

## Student's Bill's remained to Be Paid -

### Add extra biller fields if any

Add extra biller fields if any			
Bill Name	Status	Status	Close
Internet(Wi-fi)	<input checked="" type="checkbox"/>	true	<input type="button" value="X"/>
Phone	<input checked="" type="checkbox"/>	true	<input type="button" value="X"/>
House Tax	<input checked="" type="checkbox"/>	true	<input type="button" value="X"/>

## Assignment - 10

Title :- Web application using EJB

Objective :-

1. Understand about basic concepts of java beans.
2. Understand the basic functionalities of JSP, HTML.
3. Having the knowledge of JBoss Server to deploy web application.

Problem Statement :-

Design, Develop & Deploy web application using EJB.

Software Needed :-

1. Ubuntu 64 bit / windows 7
2. JDK 7 (java SE 7)
3. EJB 3.0
4. Eclipse IDE
5. JBoss Application Server

Theory :-

Java Beans :-

J2EE application contains contains the components that can be used by the client by executing the business logic. These components are known as Enterprise Java Beans (EJB).

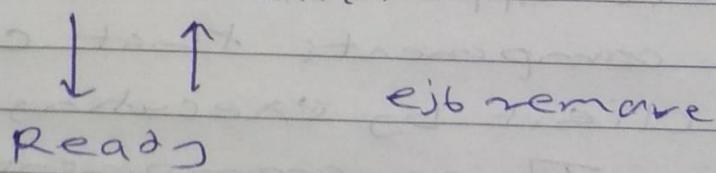
## client communication!

- State Management - Servlet development, back
- Transaction Management - Some operations like updating data
- Database connection management
- User authentication and role based authorization.
- Asynchronous messaging.
- Application server administration.

Types of Enterprise Java Beans (EJB) :-

1. Session beans
2. Entity beans
3. Message driven beans

Does not exist

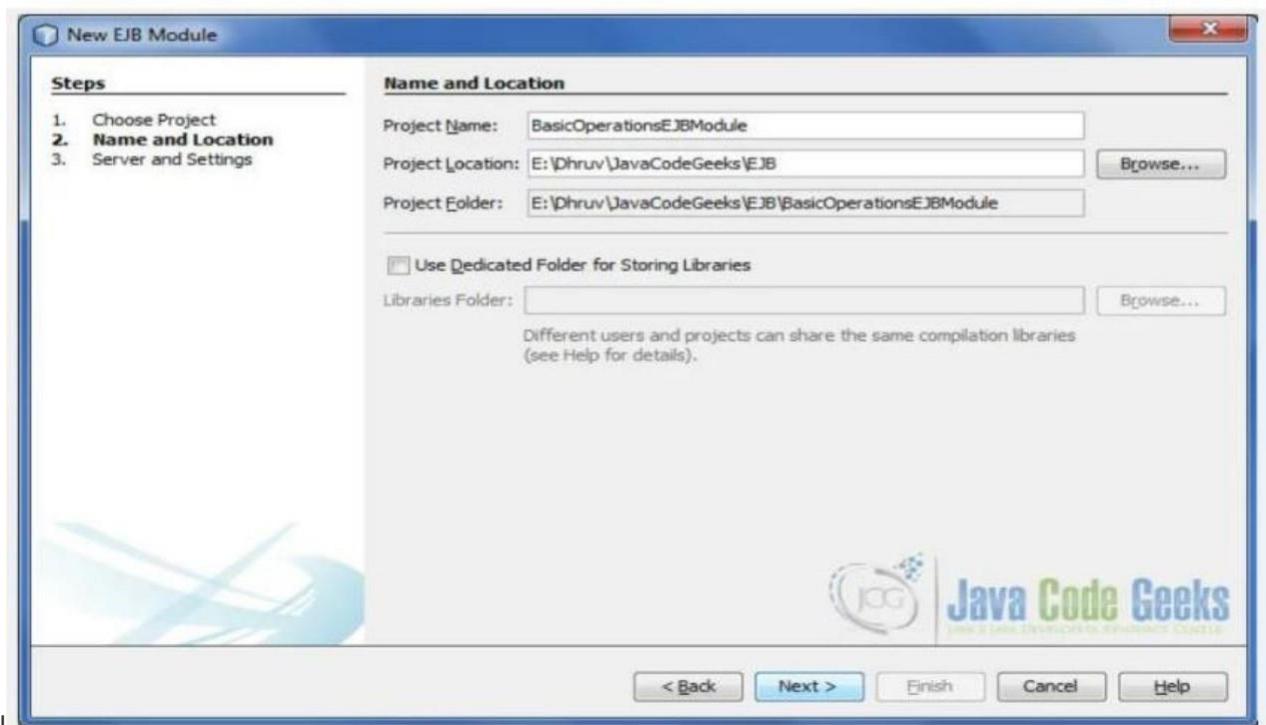
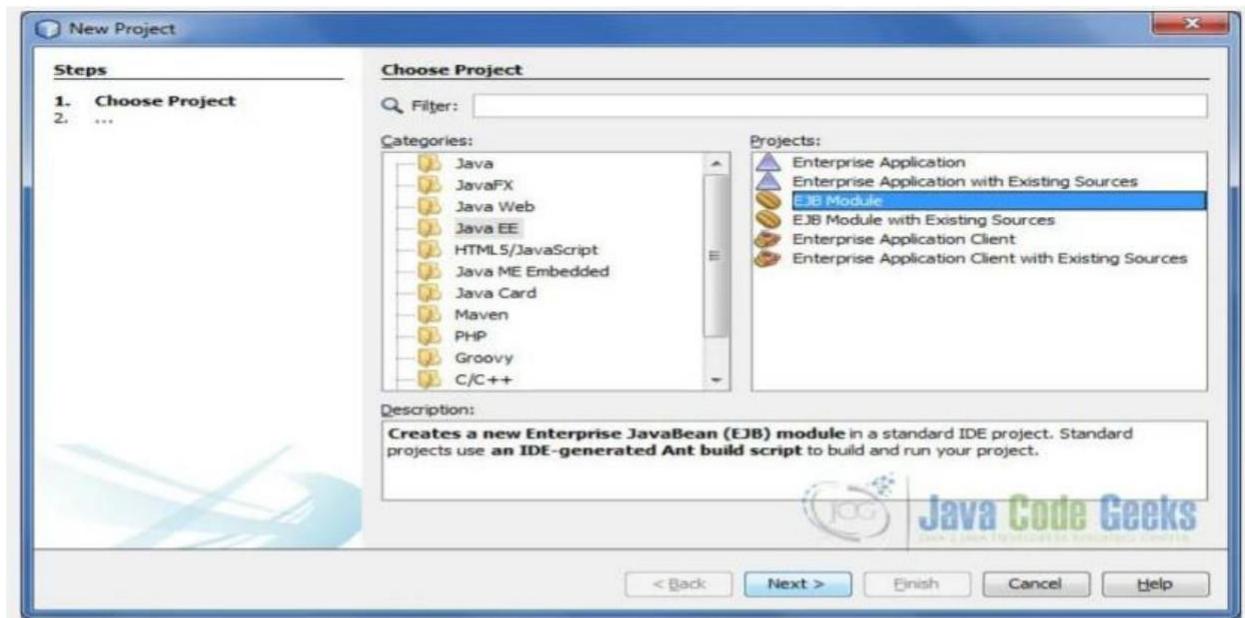


1. set session context
2. ejb create

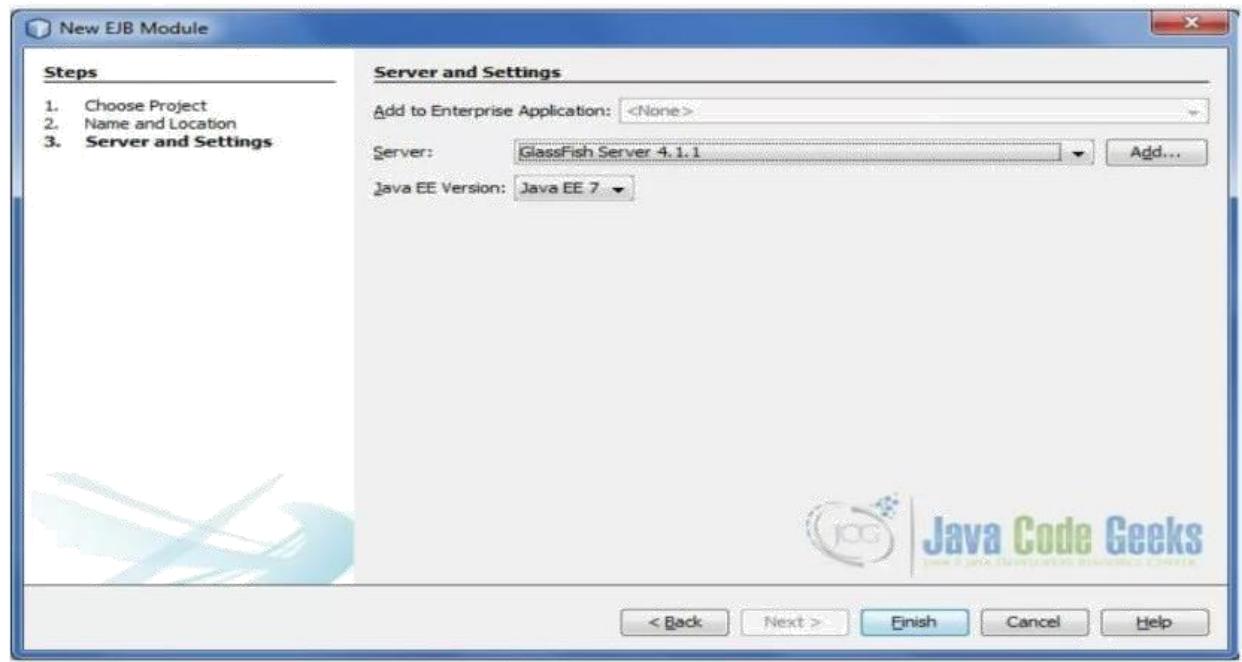
Conclusion:-

Hence, we have created a simple EJB 3 Stateless session bean and local Java application client which will call / invoke the bean to develop for performing addition of two numbers.

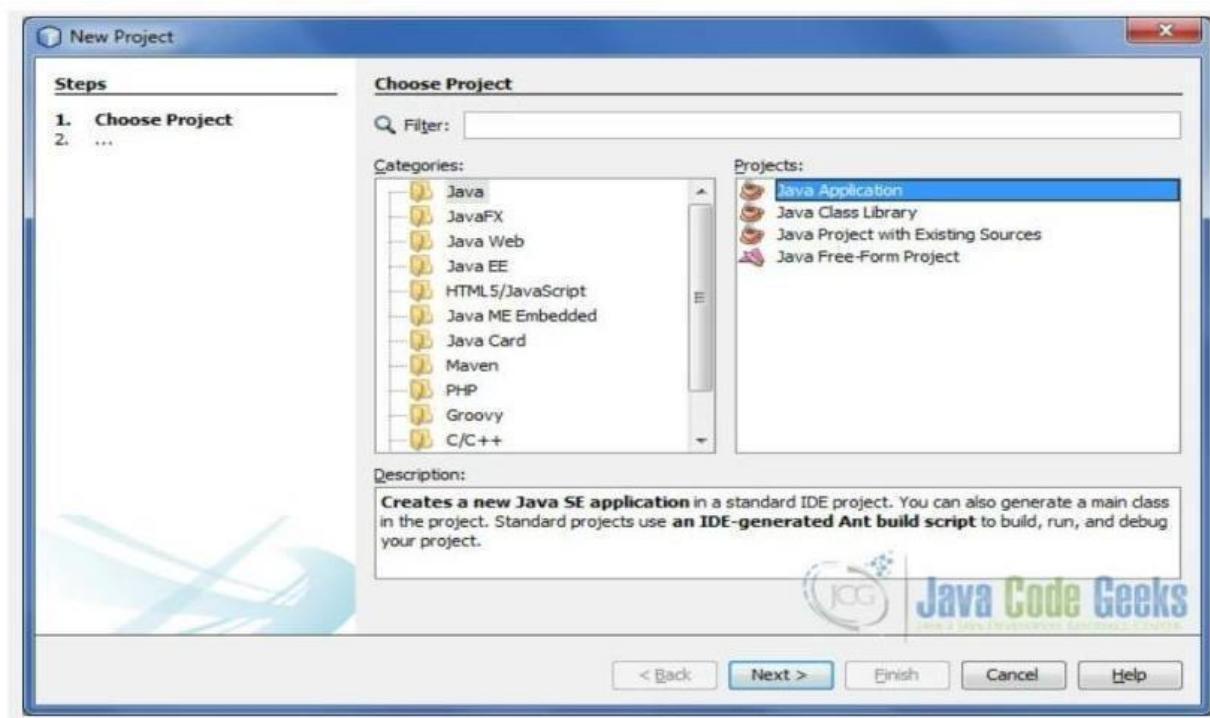
Assignment 10 : Design , develop, and deploy web application using EJB.

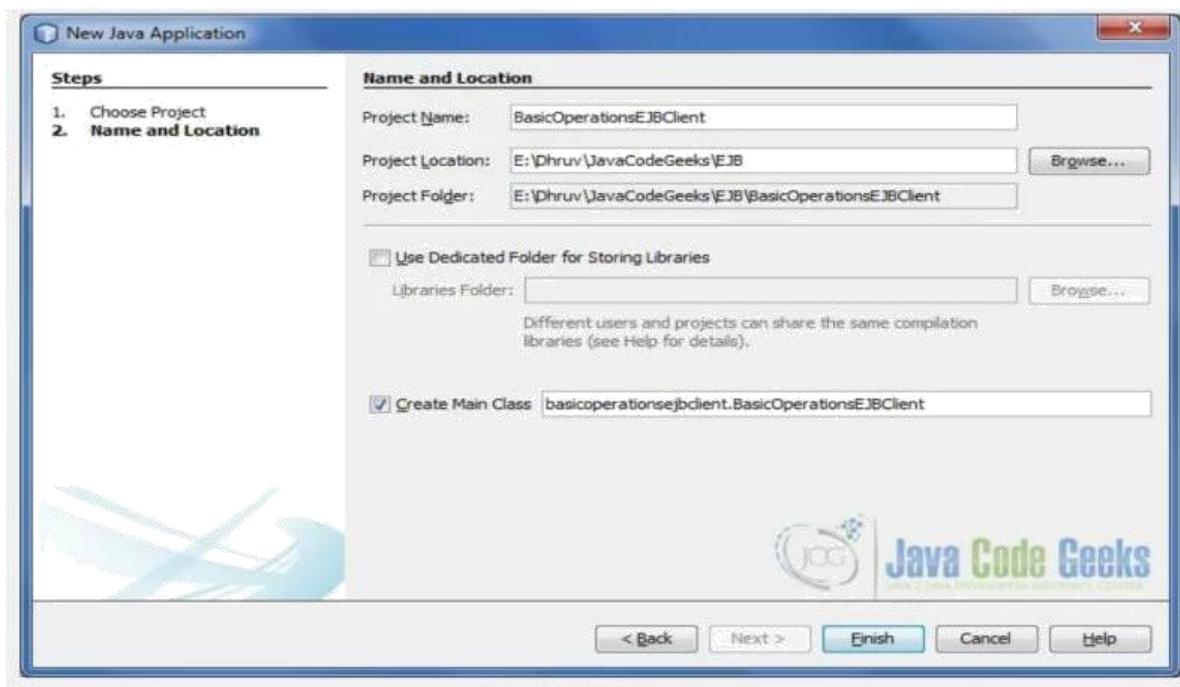


Create EJB Modul

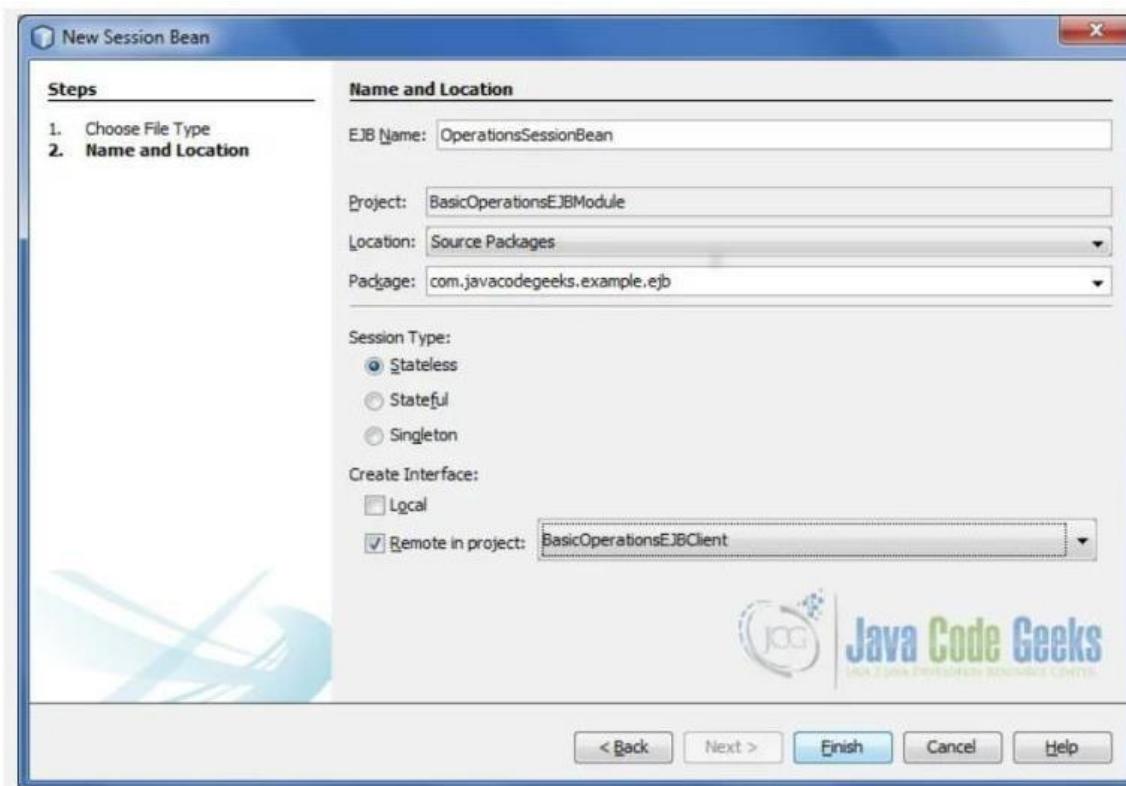


Create a new Application Class Project

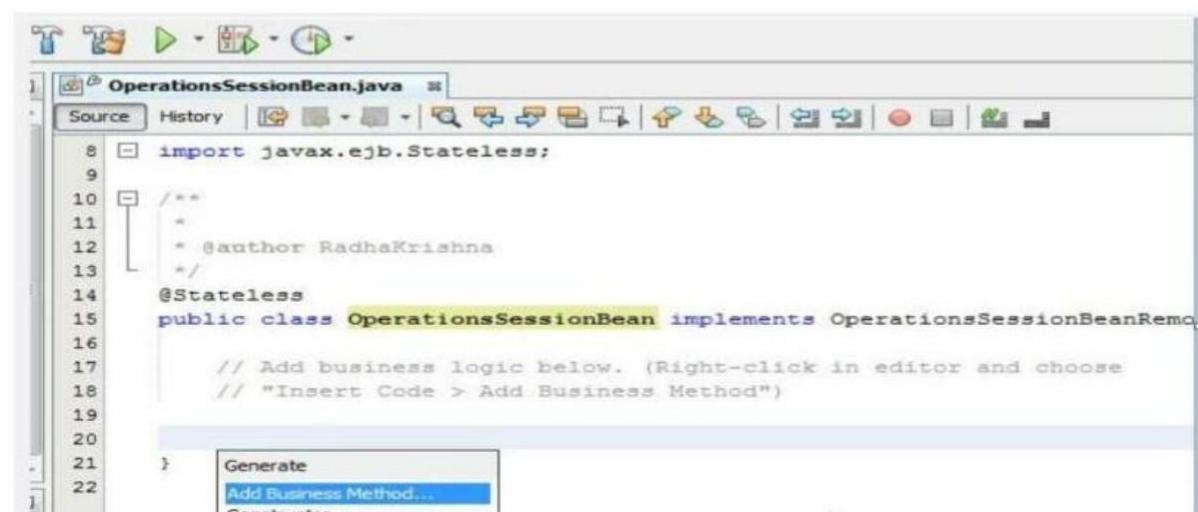
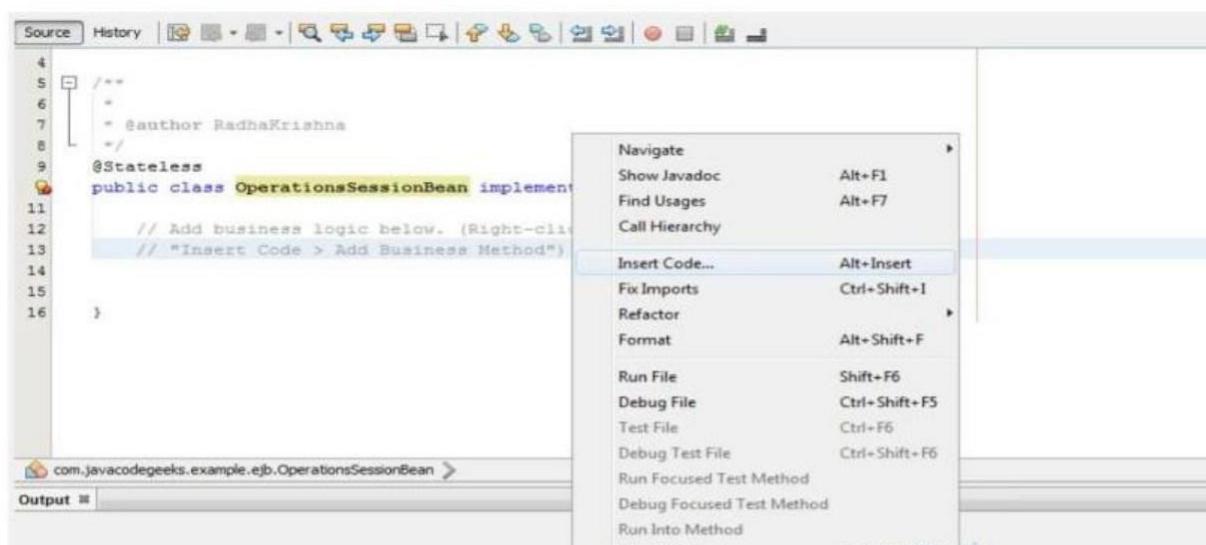


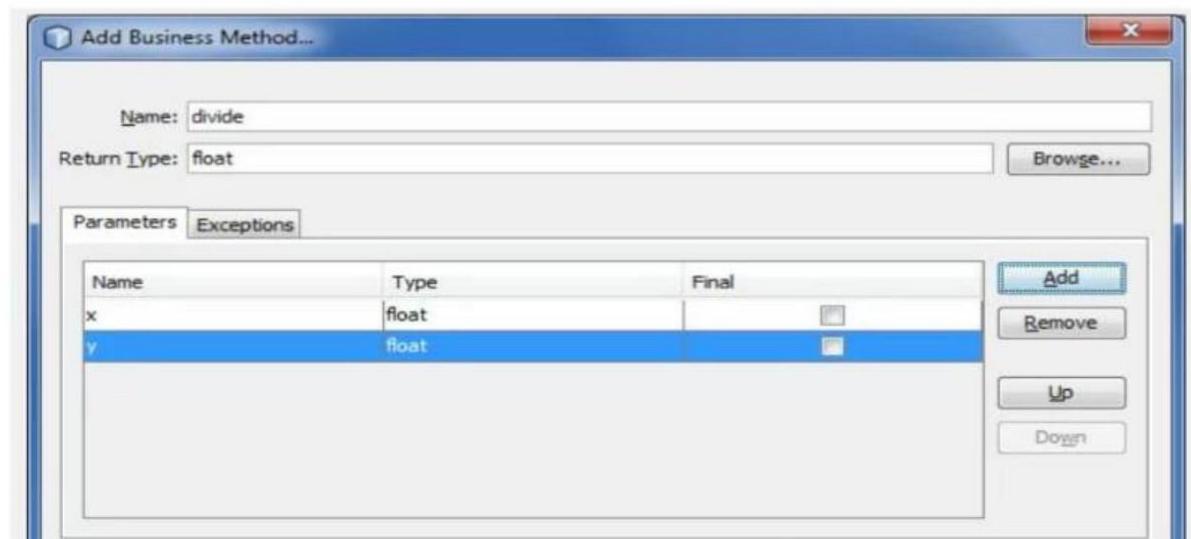
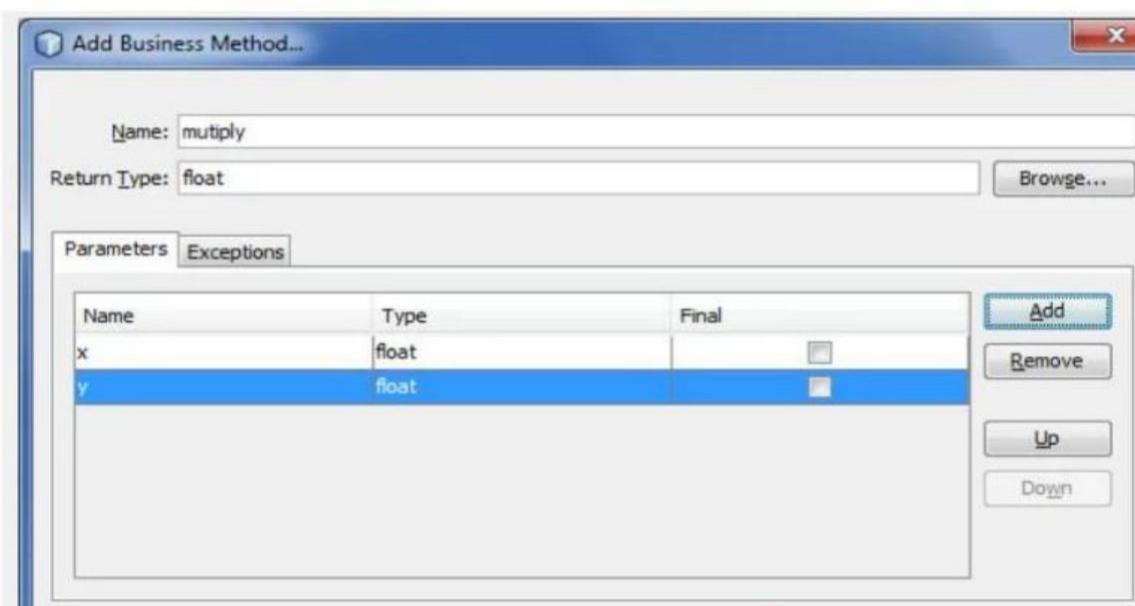
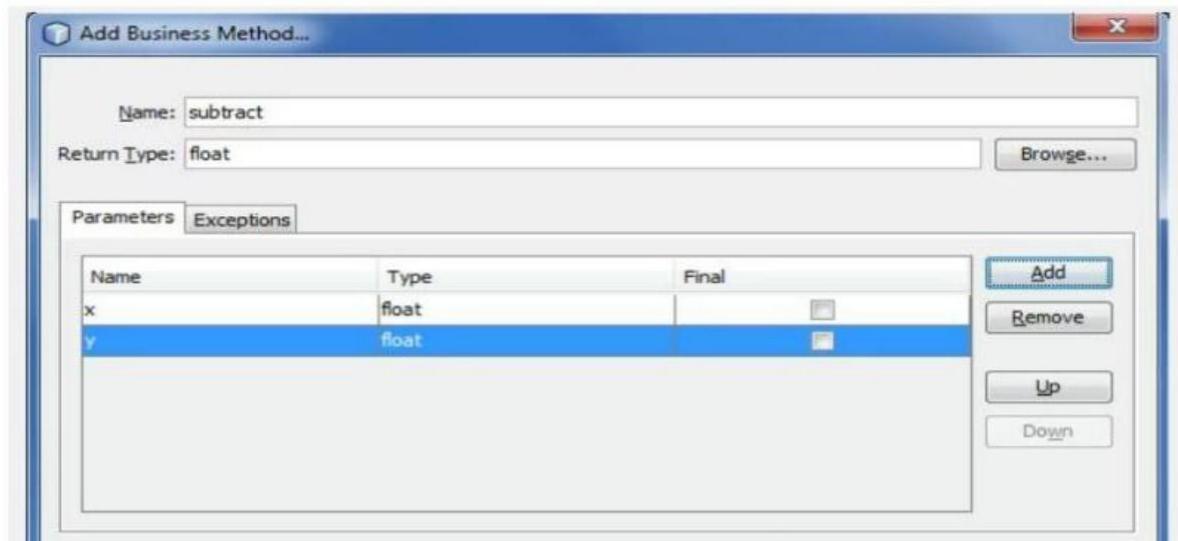


Create Session Bean



## Adding a Business Method





*OperationsSessionBean.java*

```
package com.javacodegeeks.example.ejb; import javax.ejb.Stateless;

/** * * * @author RadhaKrishna */ @Stateless public class OperationsSessionBean implements OperationsSessionBeanRemote {

    / Add business logic below. (Right-click in editor and choose / "Insert Code > Add Business Method")

    @Override
    public float add(float x, float y) {
        return x + y;
    }

    @Override
    public float subtract(float x, float y) {
        return x - y;
    }

    @Override
    public float multiply(float x, float y) {
        return x * y;
    }

    @Override
    public float divide(float x, float y) {
        return x / y;
    }
}
```

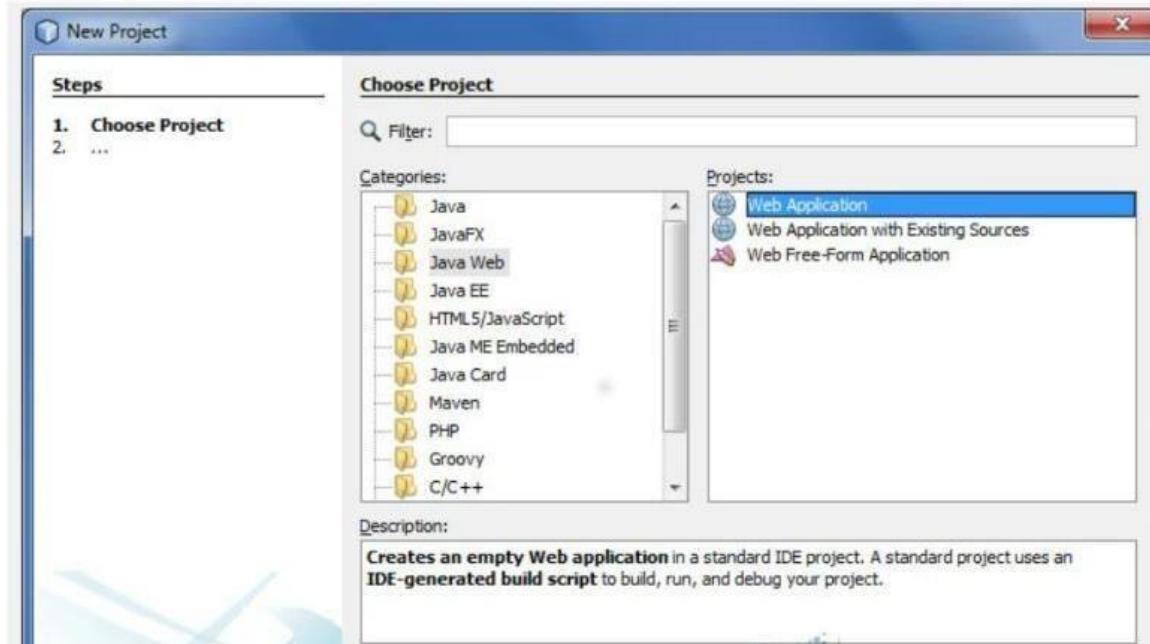
Deploy the EJB Module

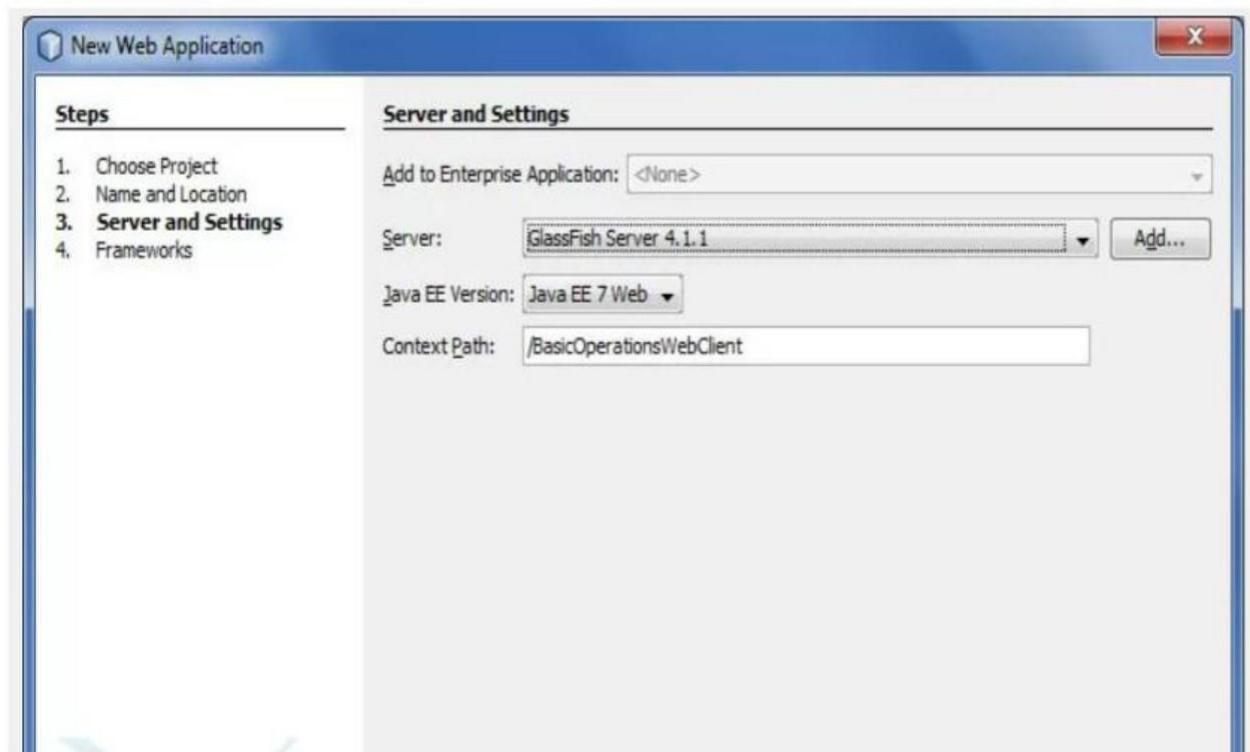
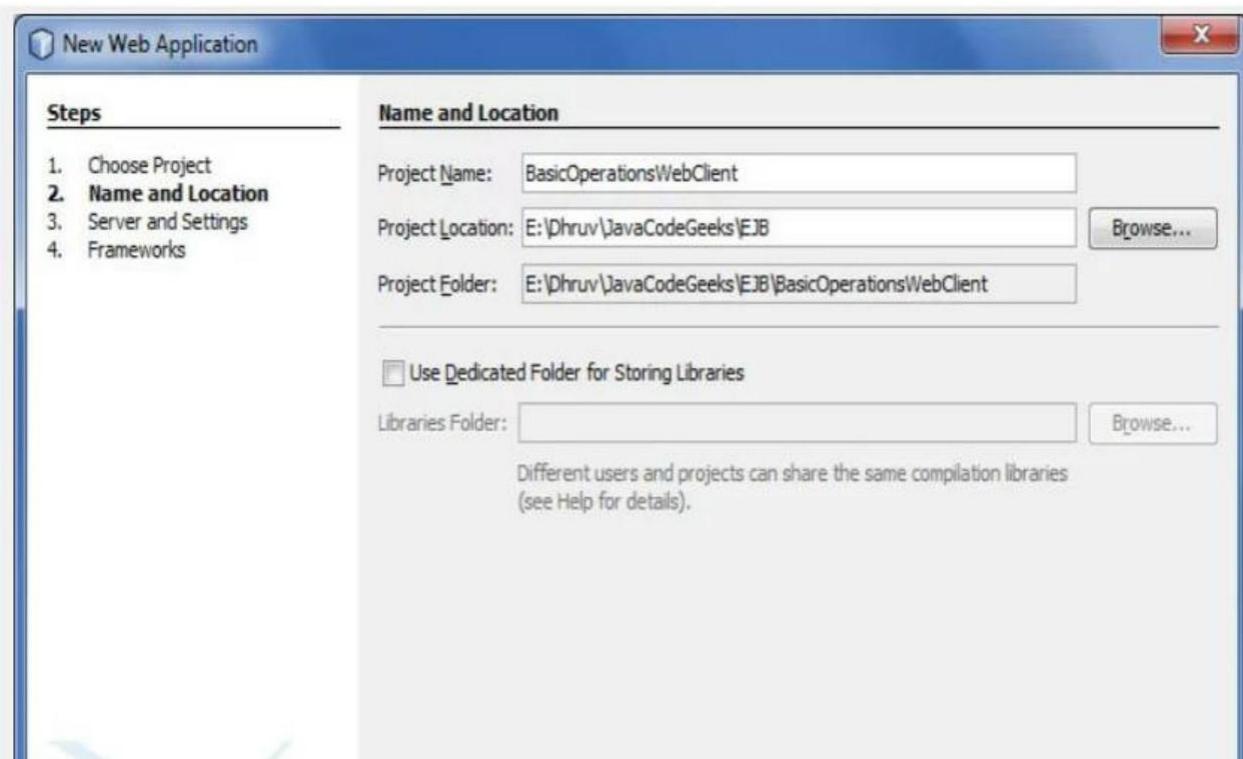
You can now build and deploy the EJB module. Right-click

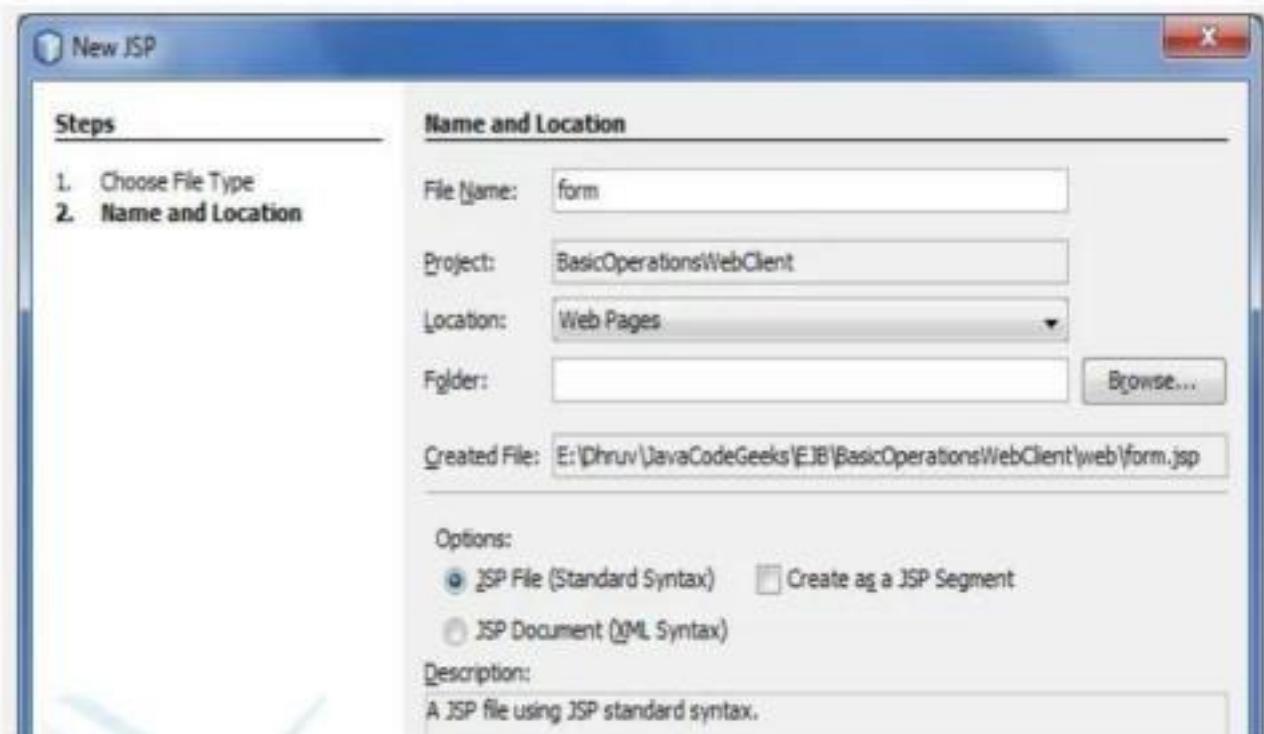
the BasicOperationsEJBModule module and choose Deploy. When you click Deploy, the IDE builds the ejb module and deploys the JAR archive to the server.

In the Services window, if you expand the Applications node of GlassFish Server you can see that BasicOperationsEJBModule was deployed.

Create a new Web Module to test EJB







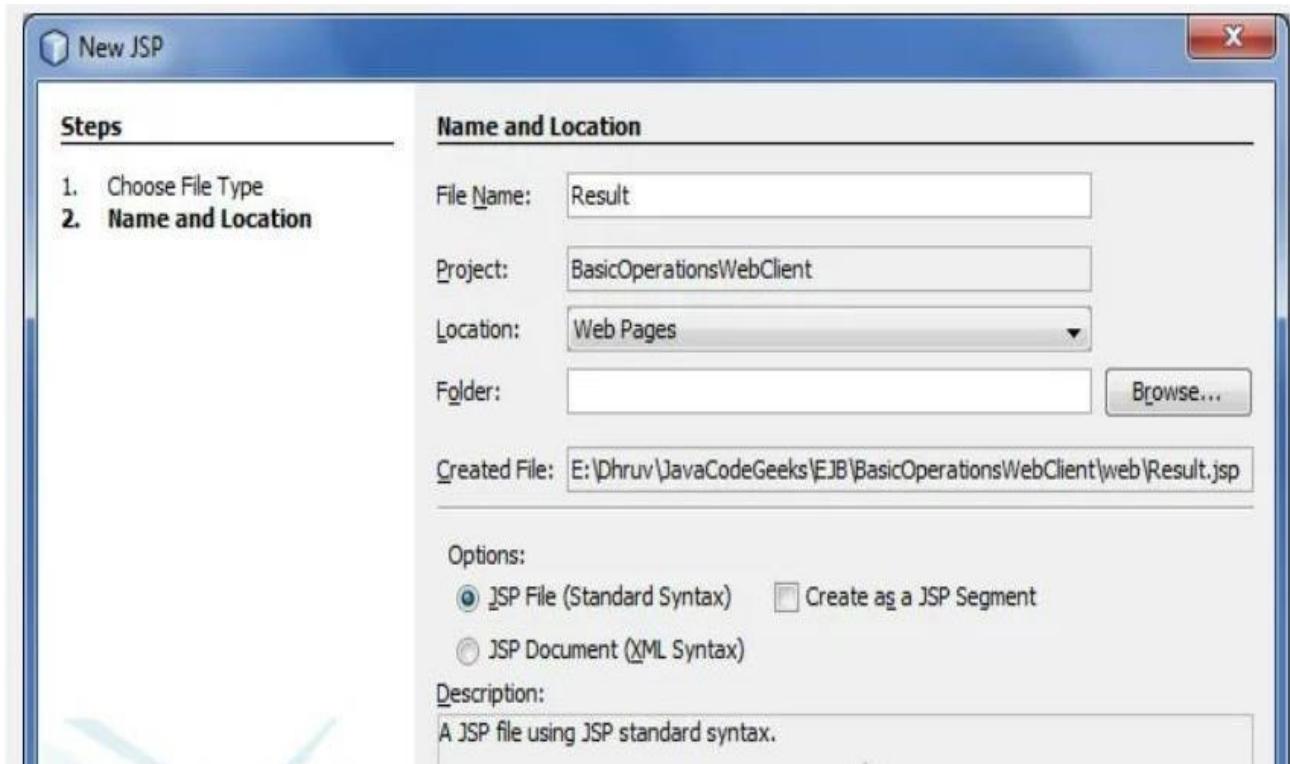
**form.jsp**

```
<html>
<head>
<title>Calculator</title>
</head>

<body bgcolor="lightgreen">
<h1>Basic Operations</h1> <hr>

<form action="Result.jsp" method="POST"> <p>Enter first value:<br>
<input type="text" name="num1" size="25"></p> <br> <p>Enter second<br>
value:<br>
<input type="text" name="num2" size="25"></p> <br>
<b>Select your choice:</b><br>
<input type="radio" name="group1" value = "add">Addition<br>
<input type="radio" name="group1" value = "sub">Subtraction<br>
<input type="radio" name="group1" value = "multi">Multiplication<br>
<input type="radio" name="group1" value = "div">Division<br> <p>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</p>
</form>
</body>
```

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



### *Result.jsp*

```
<%@ page contentType="text/html; charset=UTF-8" %>  
<%@ page import="com.javacodegeeks.example.ejb.* , javax.naming.*" %>  
  
<%!  
    private OperationsSessionBeanRemote ops = null;  
    float result = 0;  
  
    public void jsplnit() { try {  
  
        InitialContext ic = new InitialContext();  
        ops =  
            (OperationsSessionBeanRemote)ic.lookup(OperationsSessionBeanRemote.class.getName());  
  
        System.out.println("Loaded Calculator Bean");  
  
    } catch (Exception ex) { System.out.println("Error:  
        "+ex.getMessage());  
    }  
}
```

```

public void jspDestroy() {
    ops = null;
}

%>

<%
try {
    String s1 = request.getParameter("num1");
    String s2 = request.getParameter("num2");
    String s3 = request.getParameter("group1"); System.out.println(s3);

    if (s1 != null && s2 != null) {
        Float num1 = new Float(s1);
        Float num2 = new Float(s2);
        if (s3.equals("add")) {
            result = ops.add(num1.floatValue(),
                num2.floatValue()); } else if (s3.equals("sub")) {
                result = ops.subtract(num1.floatValue(), num2.floatValue());
            } else if (s3.equals("multi")) { result = ops.multiply(num1.floatValue(),
                num2.floatValue()); } else {
                result = ops.divide(num1.floatValue(), num2.floatValue());
            }
    }
}

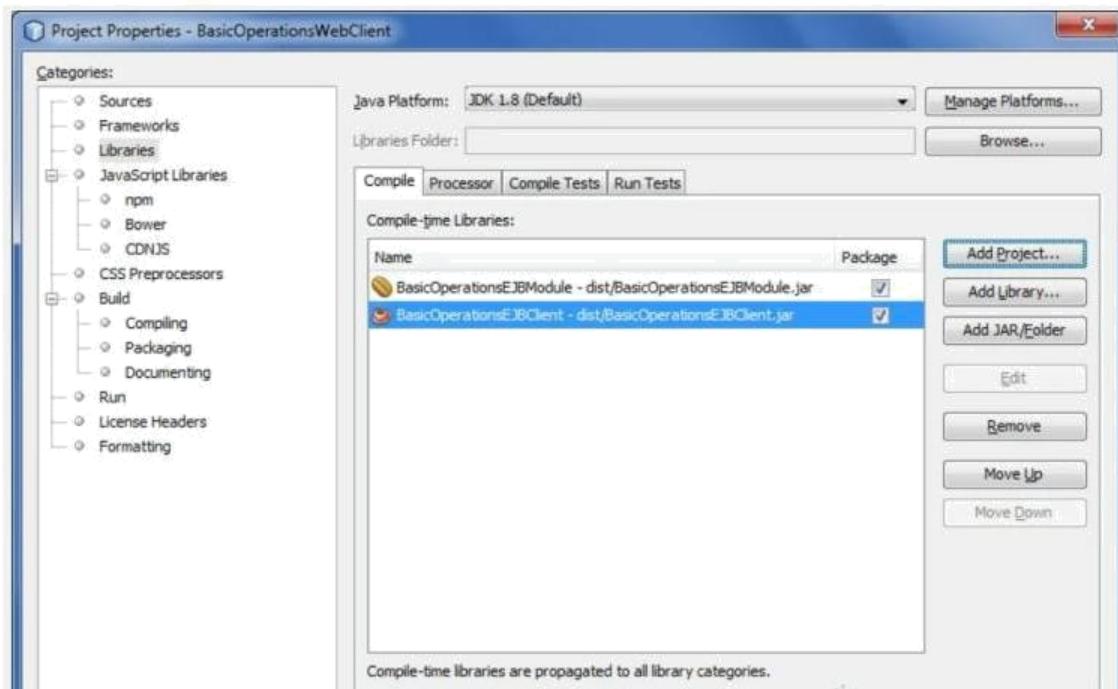
%>

<p>
<b>The result is:</b> <%= result%> <p>

<%
}
// end of try catch (Exception e) {
e.printStackTrace(); //result = "Not valid";
}

%>

```



Run the Project



http://localhost:8080/BasicOperationsWebClient/form



X

## Basic Operations

Enter first value:

Enter second value:

Select your choice:

- Addition
- Subtraction
- Multiplication
- Division

The result is: 4.0