

Assignment No 06

Page No.	
Date	

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

Program

Problem Statement

Design and develop dynamic web application using PHP and MySQL as a back-end for employee data with insert, delete, view and update operation.

Theory

1.) PHP

The PHP Hypertext Preprocessor (P) began as a little programs were written for advantage, needing an ever increasing number of individuals discovered how valuable it was Rasmus Lerdorf released the.

PHP is server side scripting dialect that is installed in HTML.

Web server - PHP will work with for all intents and purposes all Web Server programming including Microsoft Internet Information Server (IIS) however then regularly utilized is unreservedly accessible Apache server.

Database - PHP will work with for all intents and purposes all database programming including oracle and Sybase

PHP Parser :- keeping in mind that end goal to process PHP content directions a parser must be introduced to create HTML yield that can be sent to that web browser. This instructional exercise will manage you how to introduce PHP parse on your PC.

2) MySQL:

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

RDBMS Technologies

- 1) Database : A Database is a gathering of table with related information.
- 2) table : A Table is a grid with information
- 3) column : A one section contains information of one and similar kind.
- 4) Row : A line is a gathering ion related information for instance the information.

MySQL Database

- 1) MySQL is discharged under an open source permit so you don't have anything to pay to utilize it.
- 2) MySQL is a capable program in its own particular right.
- 3) MySQL utilizes a standard type of the outstanding SQL information dialect.

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.

NAME: Ujjwal Lade
ROLL NO: 06
DIV: TE-B

ASSIGNMENT NO: 6

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



Enter Name

Enter Age

Submit

```
<?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">
<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	Edit	Delete
2	Imran	18	Edit	Delete
3	Kashif	25	Edit	Delete
4	Kamran	25	Edit	Delete
5	Javed	23	Edit	Delete

```
<?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 No - 07

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

Problem Statement:

Design and develop dynamic web application using PHP, AJAX and MySQL as backend for employee data with insert and view operation.

Theory:

AJAX remains for A synchronous Javascript and XML is another procedure for making it better, speedier and more intelligent dynamic web application with the assistance of XML, HTML5 and JavaScript.

Ajax Utilize XMLHttpRequest for content CSS for introduction alongside Document object model and Javascript for dynamic substances show.

AJAX communicates with the server using XML HTTP Request object, let's try to understand the flow of Ajax or how Ajax works by ~~by~~ tr

Ajax Speaks with the server utilizing XML HTTP request question. How about we endeavour to comprehend the Stream of Ajax or how ajax functions by the picture showed beneath.

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 backend.

NAME: Ujjwal Lade

ROLL NO: 06

DIV: TE-B

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` (`id`, `name`, `address`, `gender`, `designation`, `age`, `images`) VALUES
(6, 'Barbra K. Hurley', '1241 Canis Heights Drive\r\nLos Angeles, CA 90017', 'Female', 'Service technician', 26, 'image_35.jpg'),
(7, 'Antonio J. Forbes', '403 Snyder Avenue\r\nCharlotte, NC 28208', 'Male', 'Faller', 28, 'image_36.jpg'),
(8, 'Charles D. Horst', '1636 Walnut Hill Drive\r\nCincinnati, OH 45202', 'Male', 'Financial investigator', 29, 'image_37.jpg'),
(174, 'Martha B. Tomlinson', '4005 Bird Spring Lane, Houston, TX 77002', 'Female', 'Systems programmer', 28, 'image_44.jpg'),
(162, 'Jarrod D. Jones', '3827 Bingamon Road, Garfield Heights, OH 44125',
'Male', 'Manpower development advisor', 24, 'image_3.jpg'),
(192, 'Flora Reed', '4000 Hamilton Drive Cambridge, MD 21613', 'Female', 'Machine offbearer', 27, 'image_41.jpg'),
(193, 'Donna Case', '4781 Apple Lane Peoria, IL 61602', 'Female', 'Machine operator', 26, 'image_15.jpg'),
(194, 'William Lewter', '168 Little Acres Lane Decatur, IL 62526', 'Male', 'Process engineer', 25, 'image_46.jpg'),
(195, 'Nathaniel Leger', '3200 Harley Brook Lane Meadville, PA 16335', 'Male', 'Nurse', 21, 'image_34.jpg'),
```

```
(183, 'Steve John', '108, Vile Parle, CL', 'Male', 'Software Engineer', 29, 'image_47.jpg'),  
(186, 'Louis C. Charmis', '1462 Juniper Drive\r\nBreckenridge, MI 48612', 'Male', 'Mental health  
counselor', 30, ''),  
(200, 'June Barnard', '4465 Woodland Terrace Folsom, CA 95630', 'Female', 'Fishing vessel operator',  
24, '');
```

- 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>Gender</td>
<td>Designation</td>
<td>Age</td>
<td>View</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('address', $('#address').val()); form_data.append('gender', $('#gender').val());  
form_data.append('designation', $('#designation').val()); form_data.append('age', $('#age').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 :

Employee Details							Actions	
Name	Address	Description	Age	View	Update	Delete		
John Doe	123 Main St	Manager	35	<button>View</button>	<button>Update</button>	<button>Delete</button>		
Jane Smith	456 Elm St	Developer	28	<button>View</button>	<button>Update</button>	<button>Delete</button>		
Showing 1 to 2 of 2 total records								
Previous				1	2	Next	Final	

Assignment 08

Title: Design and develop my web application using structs framework.

Problem Statement:

Create a login module for the web application using Structs frameworks.

Theory:-

framework plays a vital role in industries for manageable and well designed. application development the core of the structs framework is flexible control layer based on standard technologies like java servlet Java Beans Resource bundles and JME as well as various Jakarta communication commons packages thus encourages application architecture based on the model.

- Model - view controller Architecture
 - model view controller is a way to build application that promotes complete separation between business logic and presentation. It a not specifies to web application or JSP or JEEF the view is the user interface the screens that the end user of the application actually sees and interacts

with. In a J2EE web application views as JSP files for collecting user input you will have a JSP that generates an HTML page that contains one or more HTML form.

Structs

— Structs is a framework that advances the utilization of the model-view-controller engineering for planning substantial scale application. The structure incorporates an arrangement of custom label libraries and their related Java classes alongside different utility classes.

Structs tags

Common Attributes.

Almost All tags provided by the Structs framework use the following Attributes:

Attributes — used for

Id — the name of bean for temporary used by the tag

name — the name of pre-existing bean for use with the tag

property — the property of the bean named in the attribute for use with the tag

Scope — the scope to search for the bean named in the name attribute

Creating HTML forms.

Frequently data should be gathered from a client and handled without the capacity to gather client input, a web application would be futile. So as to get the clients data, a html shape is utilized, client information can originate from a few gadgets : for example contents fields, content boxes of check boxes or pop menus and radio catches.

Conclusion

Hence we have successfully tested the streets Framework and tested the result.

NAME: Ujjwal Lade

ROLL NO: 06

DIV: TE-B

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">
<struts-config>
<form-beans>
<form-bean name="loginForm" type="com.example.javawebtutor.form.LoginForm"/>
</form-beans>
<action-mappings>
<action name="loginForm" path="/login" type="com.example.javawebtutor.action.LoginAction"
scope="request" input="/login.jsp">
<forward name="failure" path="/login.jsp" redirect="true"/>
<forward name="success" path="/success.jsp" redirect="true"/> </action>
</action-mappings>
</struts-config>
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>
```

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 No - 09

Page No. _____
Date. _____

Title : Design and develop any web application using Angular JS.

Problem statement . . .

Create an application for Bill Payment Record using AngularJS

Theory -

AngularJS is an open Source web application framework. It was initially created in 2009 by Misko Hevery and Adam Abrons.

- General features

- 1) Angular JS is a productive System that can make Rich Internet Applications
- 2) Angular JS gives designer a choice to compose customer side
- 3) Application written in Angular JS are cross programs aggregable.

Core features:

① Data authorization

② Scope

③ Controller

④ Services

⑤ Filters

⑥ Template

⑦ Directives

⑧ Routing

⑨ Model view whatever

⑩ Deep Linking

⑪ Dependency Injection

Advantages

- ① It gives the ability to make Single Page Application in a spotless and variable way.
- ② Angular.js code is unit testable
- ③ Angular.js gives reusable segment

Model view Controller

MVC as it is called is a product configuration design for creating Web application.

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 a part of the information to the client.

Controller - It is a product code that controls the connections between the model and view.

AngularJS is a MVC Based Structure,

- An AngularJS application comprises of following three essential parts
 - ng-app- This directive defines and

Link and AngularJS application to HTML

- **ng-model** - This directive binds the values of AngularJS application data to HTML input method.
- **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 model structure and its use in advanced web app programming is studied.

NAME: Ujjwal Lade

ROLL NO: 06

DIV: TE-B

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 -<br/>

<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:

Sanchita's Bills to Be Paid - 3

	+Add+
Bill Name	Status
Electricity	<input type="checkbox"/>
Internet(Wi-fi)	<input type="checkbox"/>
Parking Charges	<input type="checkbox"/>
Phone	<input checked="" type="checkbox"/>
House Tax	<input checked="" type="checkbox"/>

Assignment No 10

Page No.	
Date	

Title - Web Application using EJB

Problem Statement

Design Develop & Deploy web application using EJB.

Theory:

Java Beans

J2EE application container the components that can be used by the clients for executing the business logic these components are known as enterprise Java Beans (EJB). EJB 3.0 is being a large shift from EJB 2.0 and makes development of EJB based applications relatively easy.

Features of EJBs.

- Client Communication

the client which is open a user interface must be able to call the methods

- store management you will recall our discussion on this topic.

- Transaction management

- Database connection management

- User Authentication and Role base functions ratio

- Asynchronous messaging.

- Application fe Server Administration

Types of Enterprises Java Beans.

① Session Beans

- Session Beans are intended to allow the application author to easily implement portion of application code in middleware to simplify access to this code.
 - It is not stored
 - It is not persistent

② Stateless Session Bean

- The client invoked the create method
- The EJB container
- instantiates the bean.
- The Bean is ready
- while in the ready state
- client may invoked a business method

Conclusion -

Hence we have created a simple EJB of Stateless Session bean and local Java application client which will call invoke the develop for performing addition of two Numbers

NAME: Ujjwal Lade

ROLL NO: 06

DIV: TE-B

Assignment 10

Design , develop, and deploy web application using EJB.

Create EJB Module



