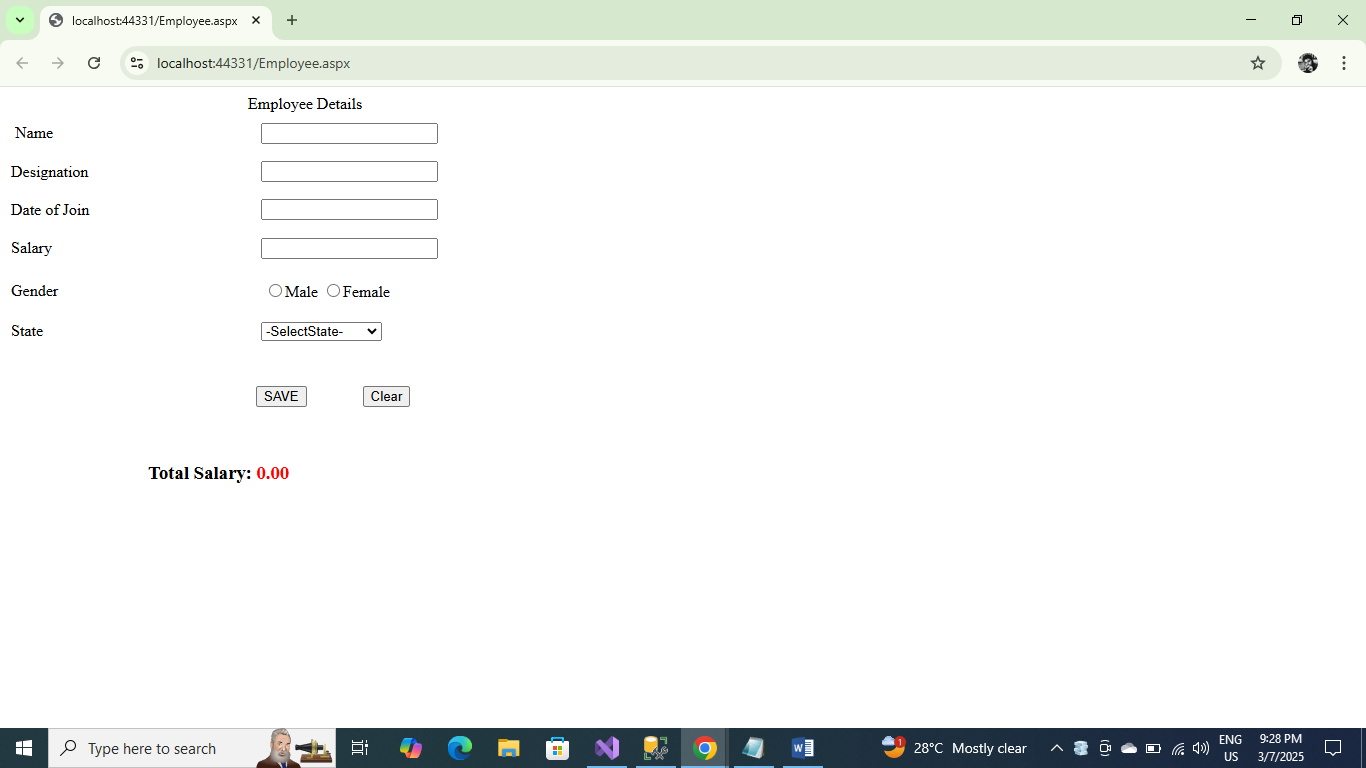
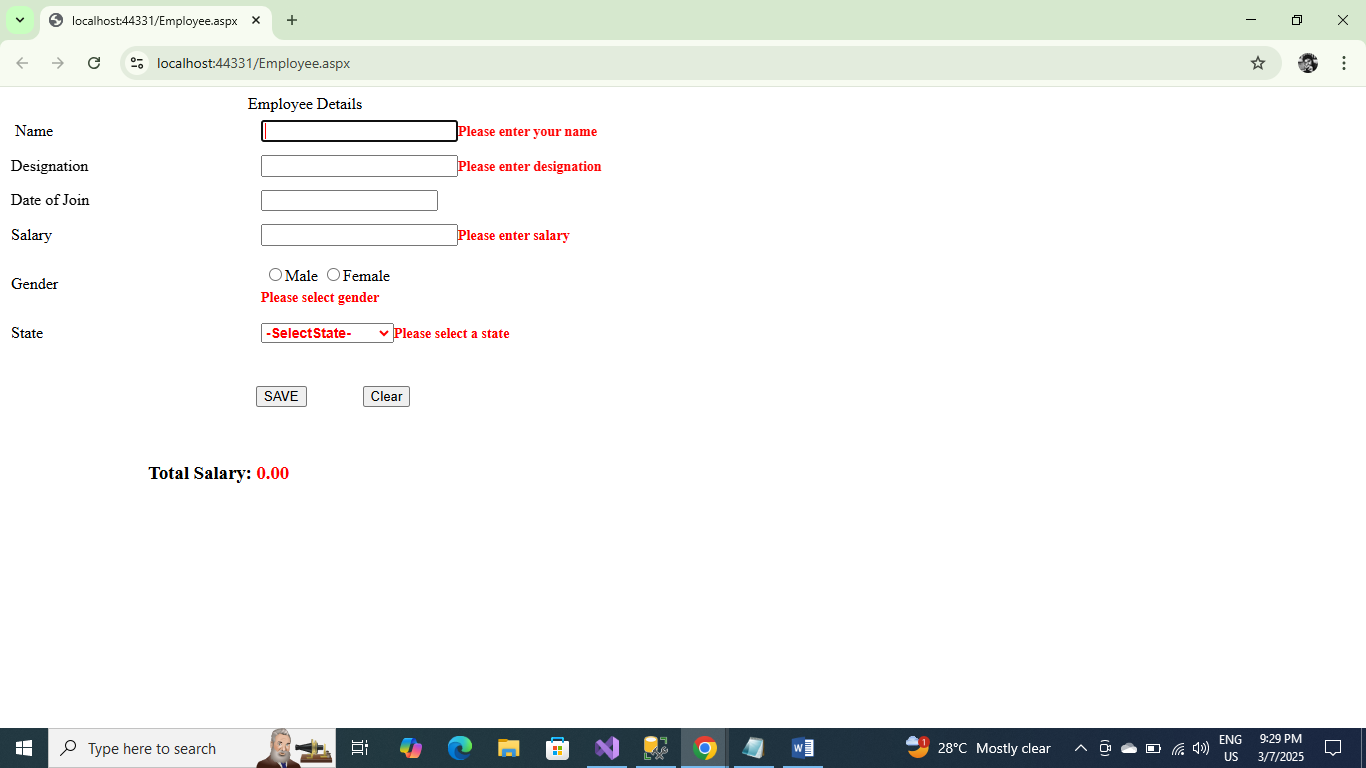
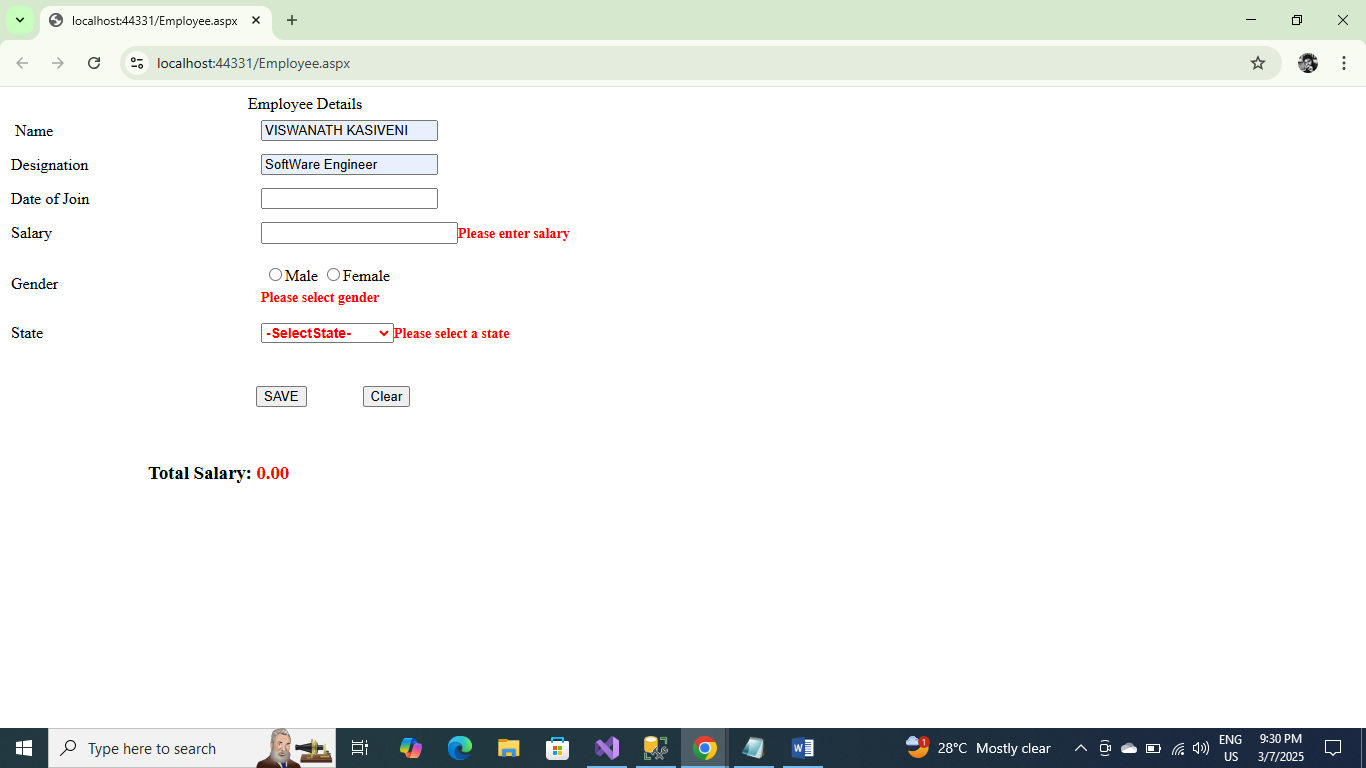
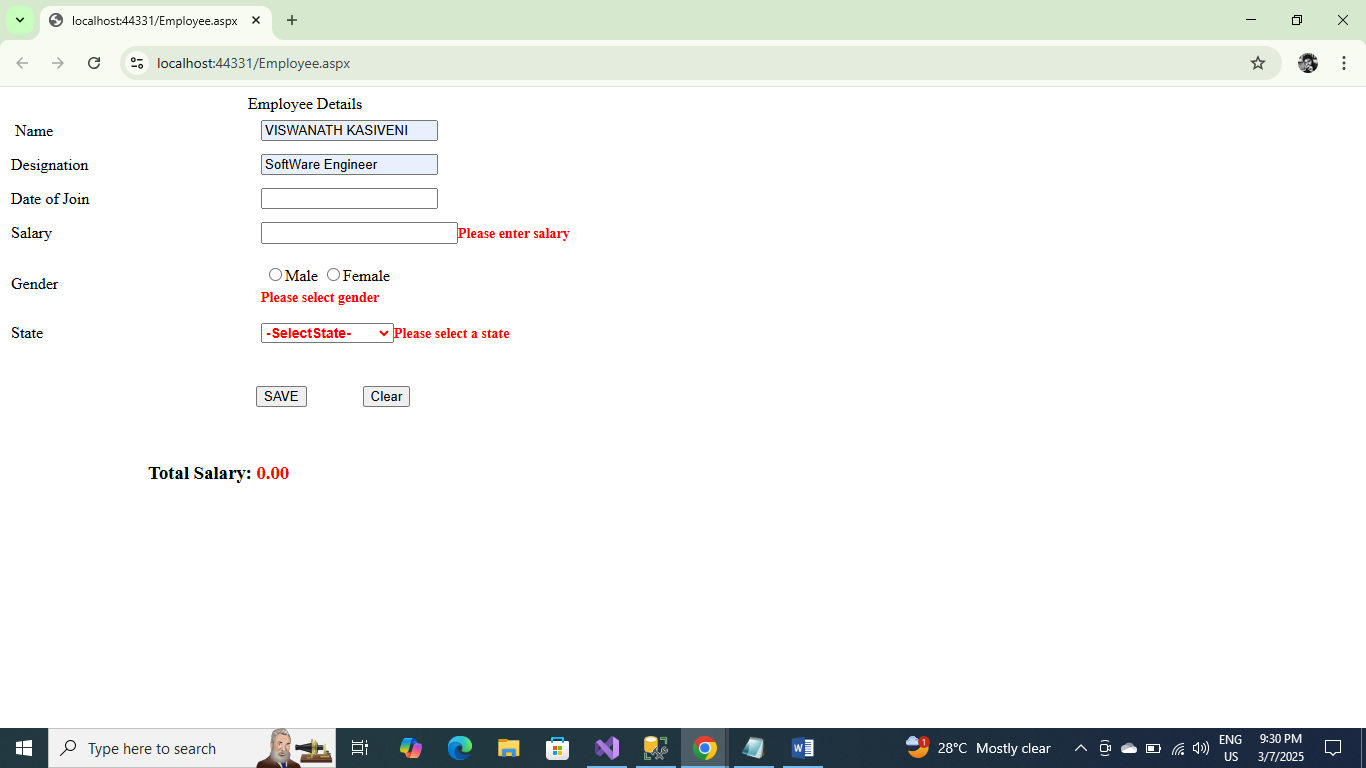
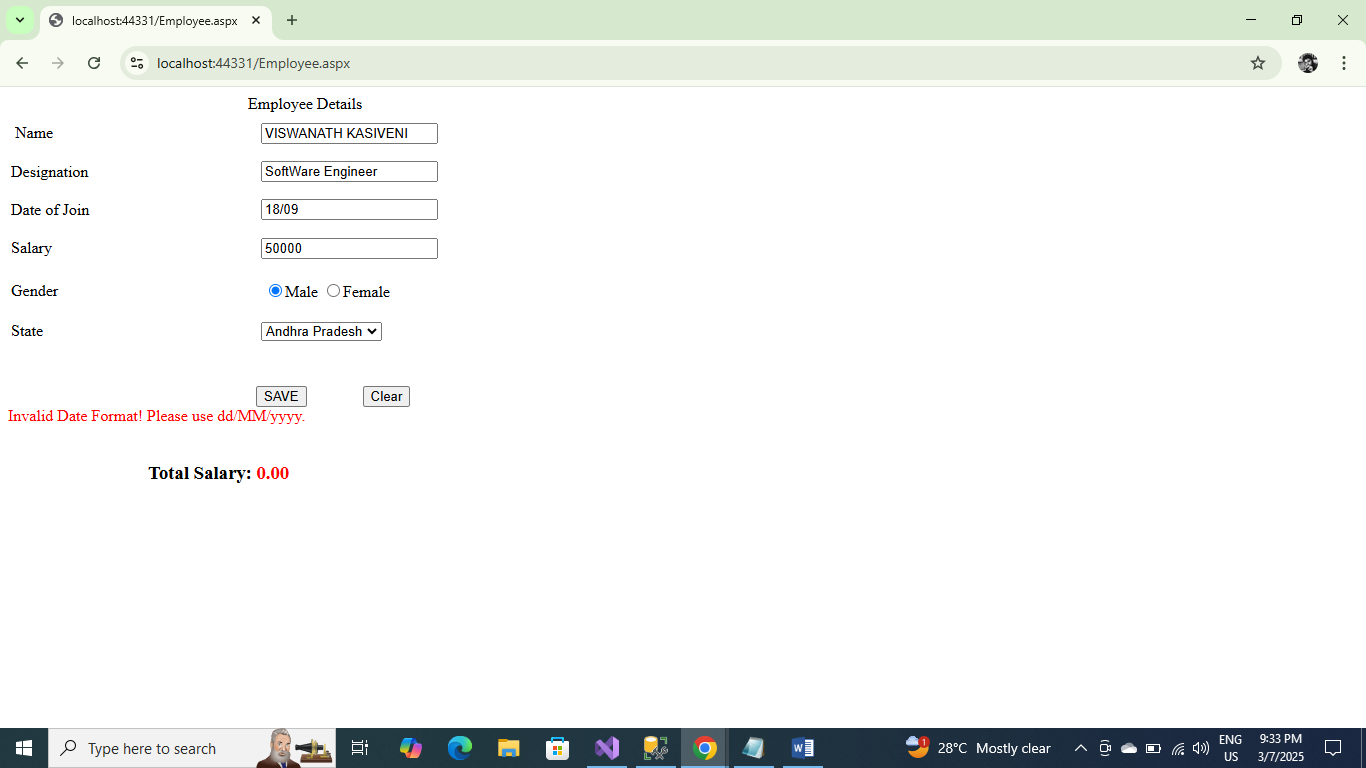
**Employee Details**

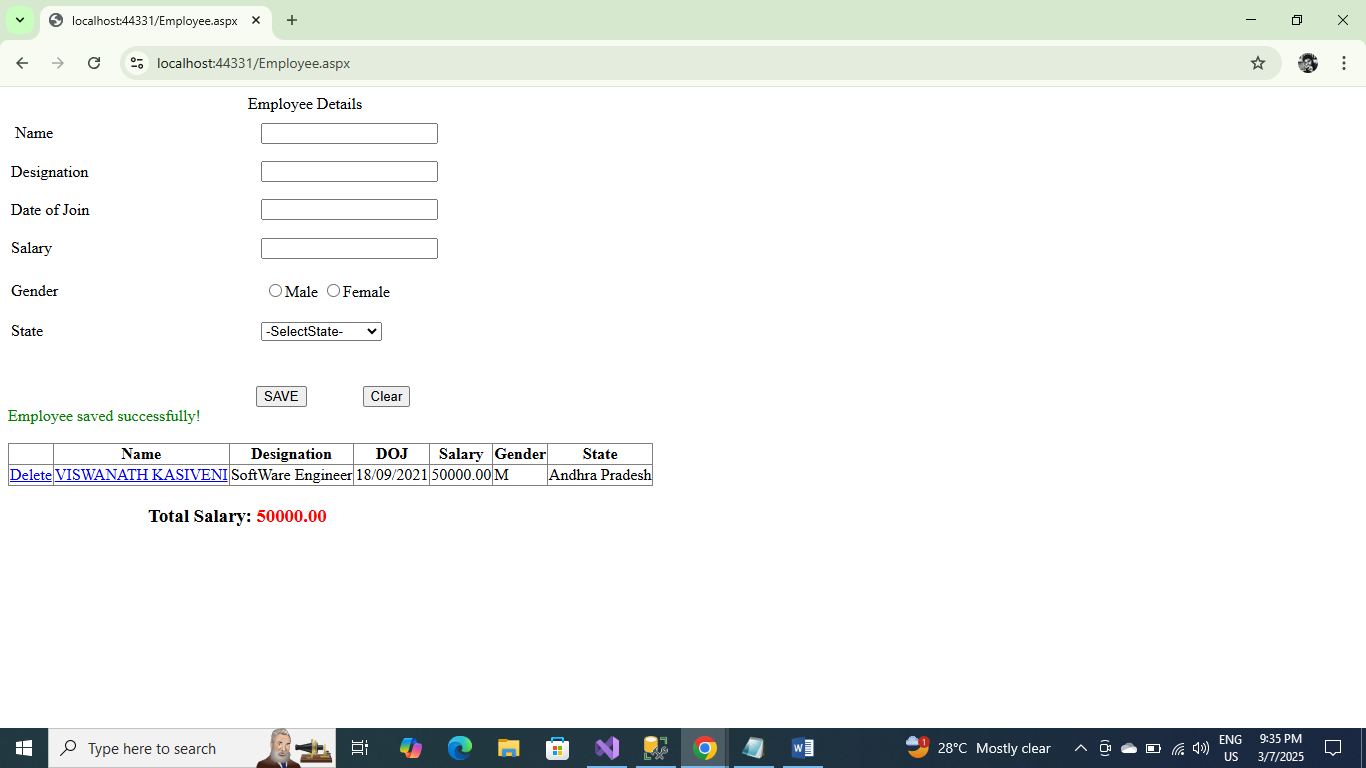
1.When the application runs for the first time, it displays.. 

2. Display jQuery validation ..

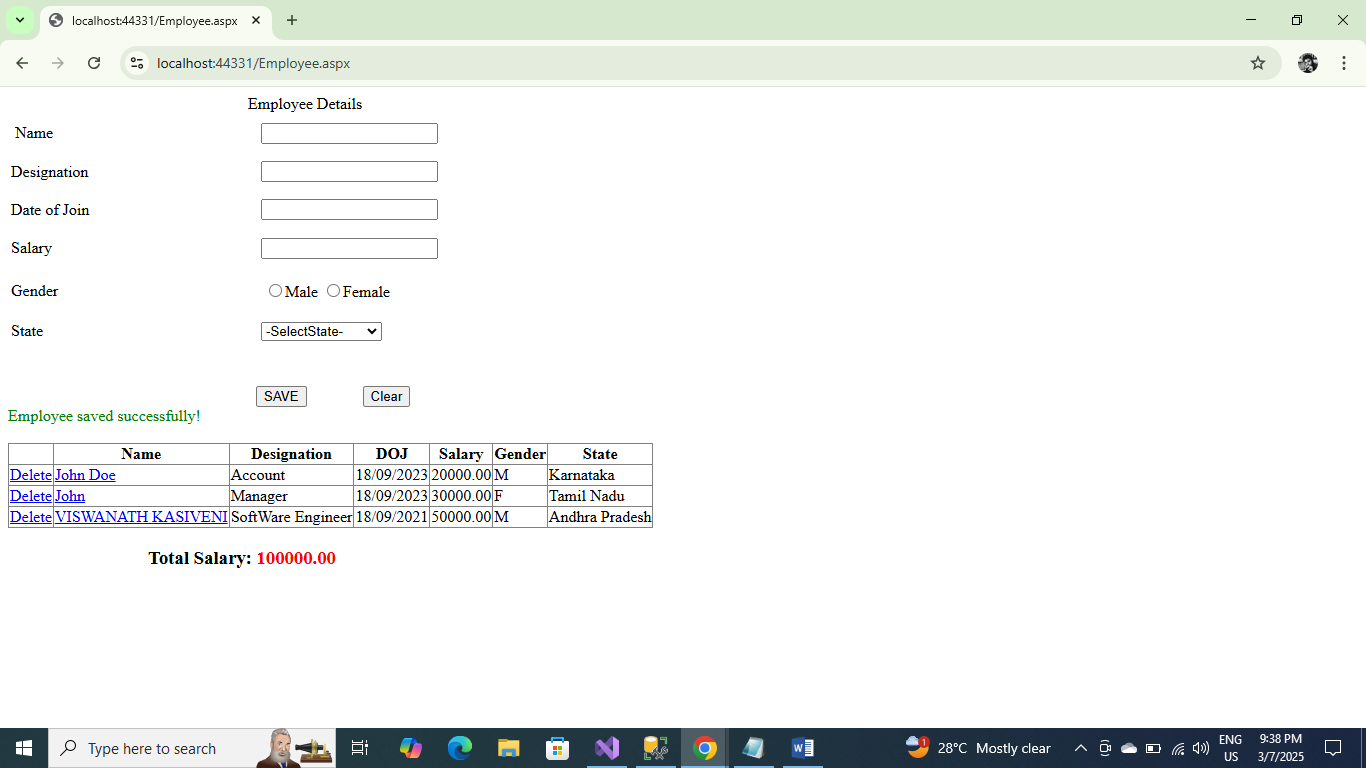
3. Remove validation one by one as the user enters input. 

4. Display salary validation only when numbers are entered. 

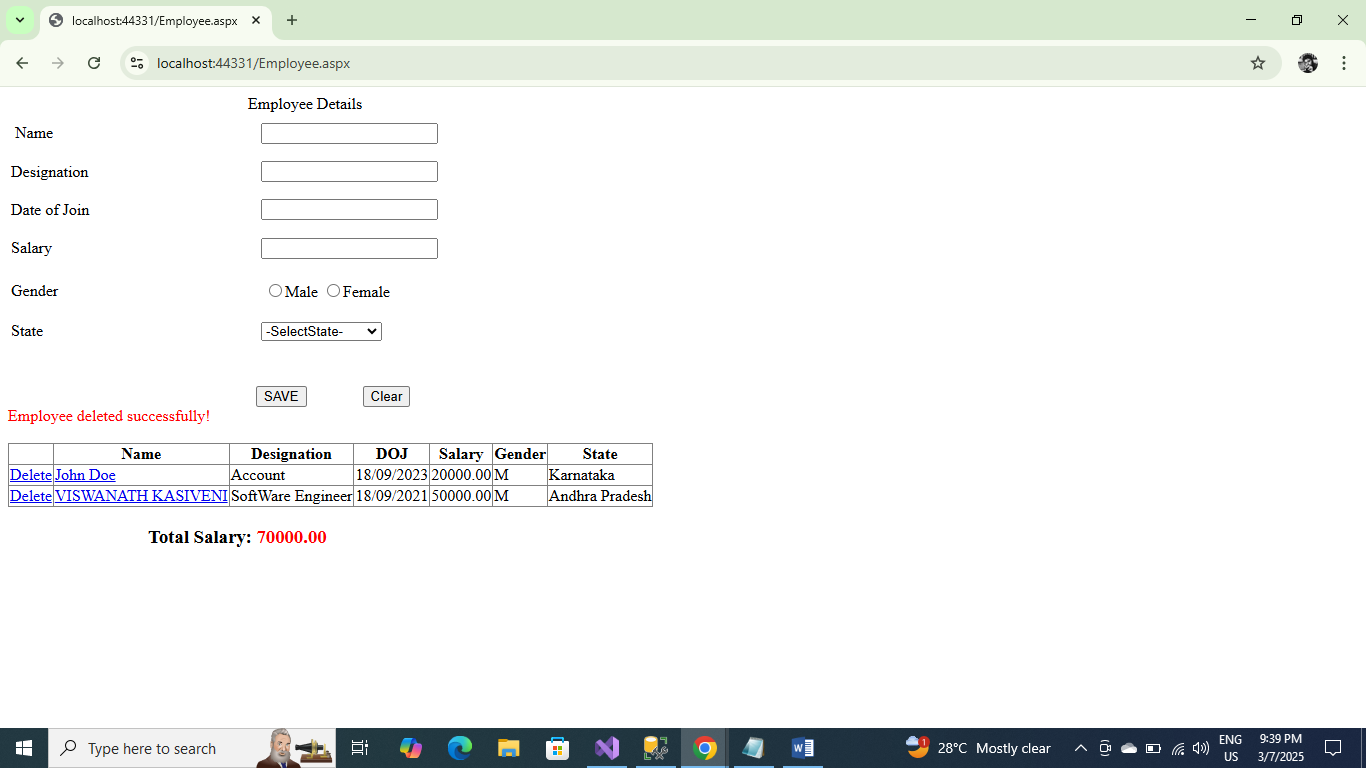
5. Display validation when the user enters an incorrect date format. 

6. Once the details are saved, display a message, bind the data to the GridView, update the sum of salaries, and clear all field values.

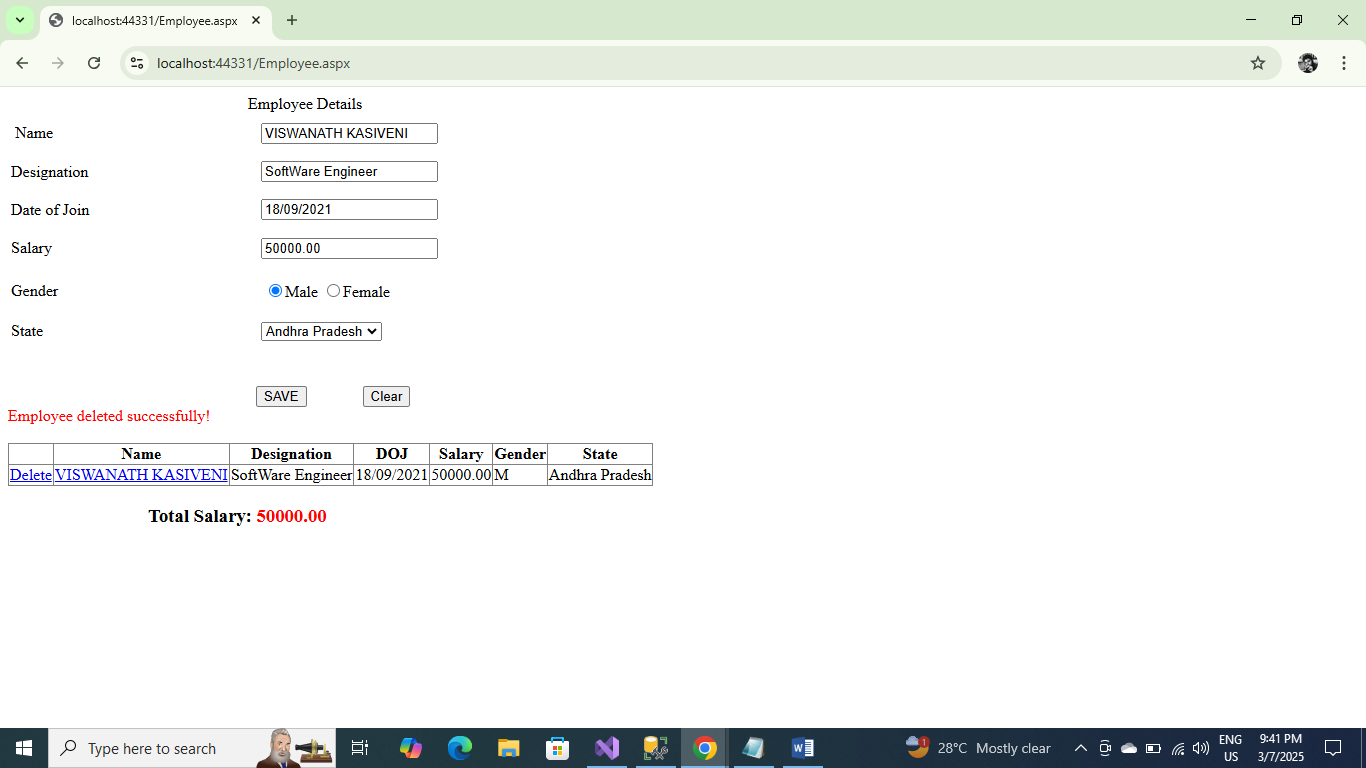
7. Calculate the sum of salaries.



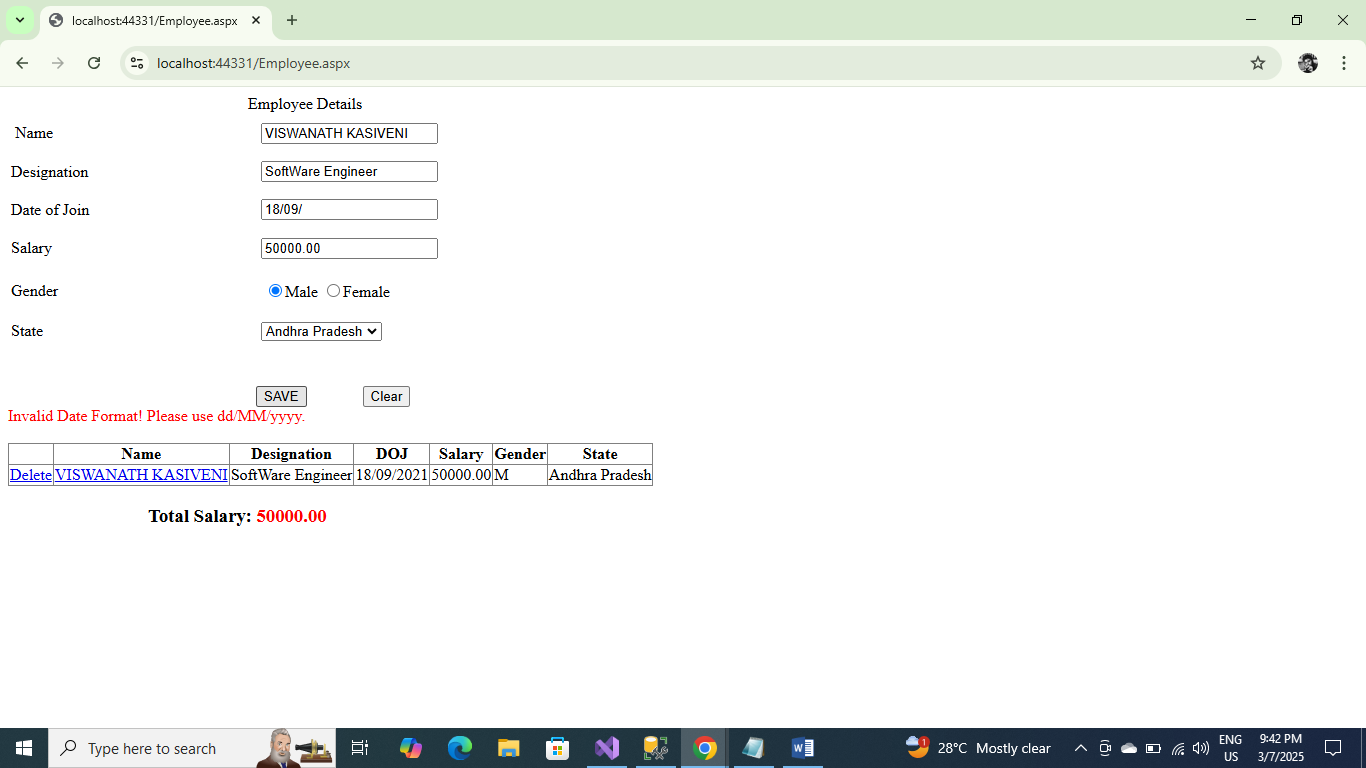
8. While deleting a record, the selected record will be deleted.



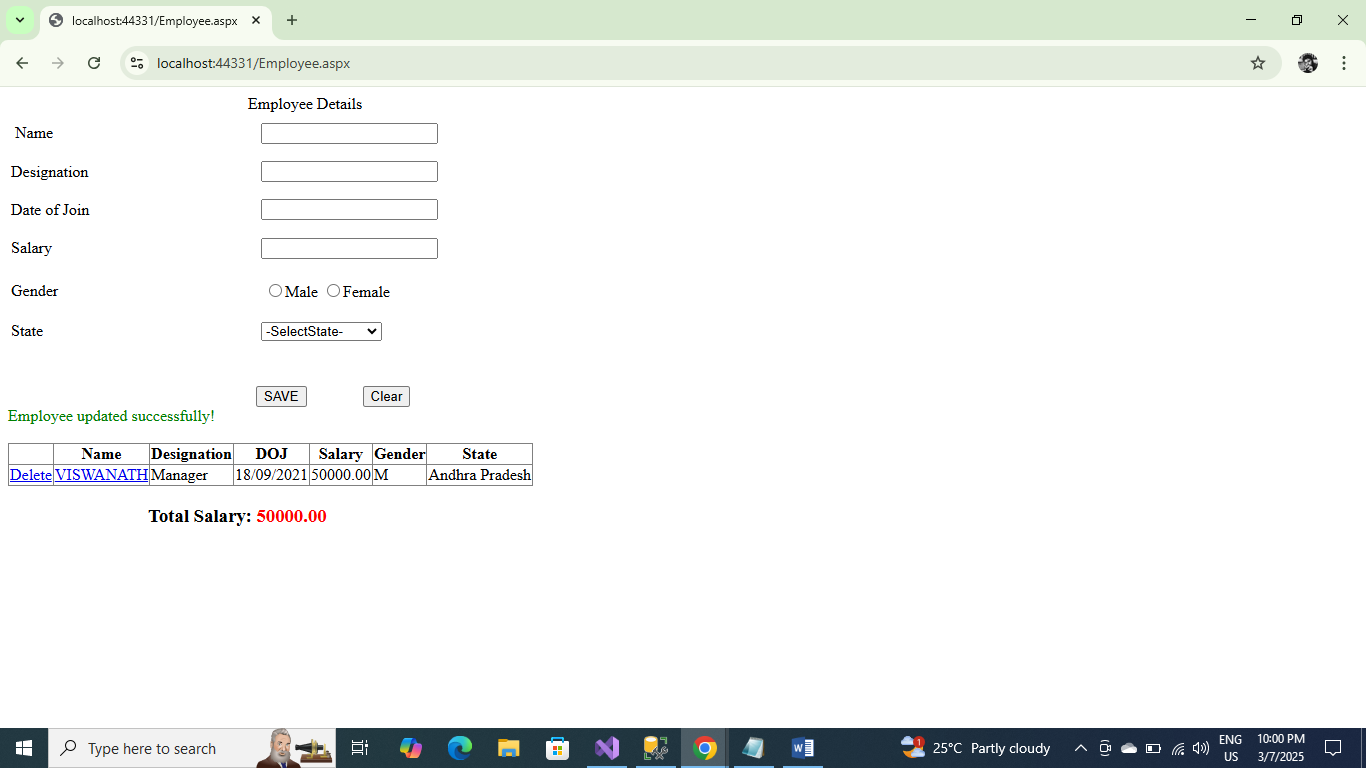
9. When the user clicks on a Name, the details will be displayed above based on the field values for editing.



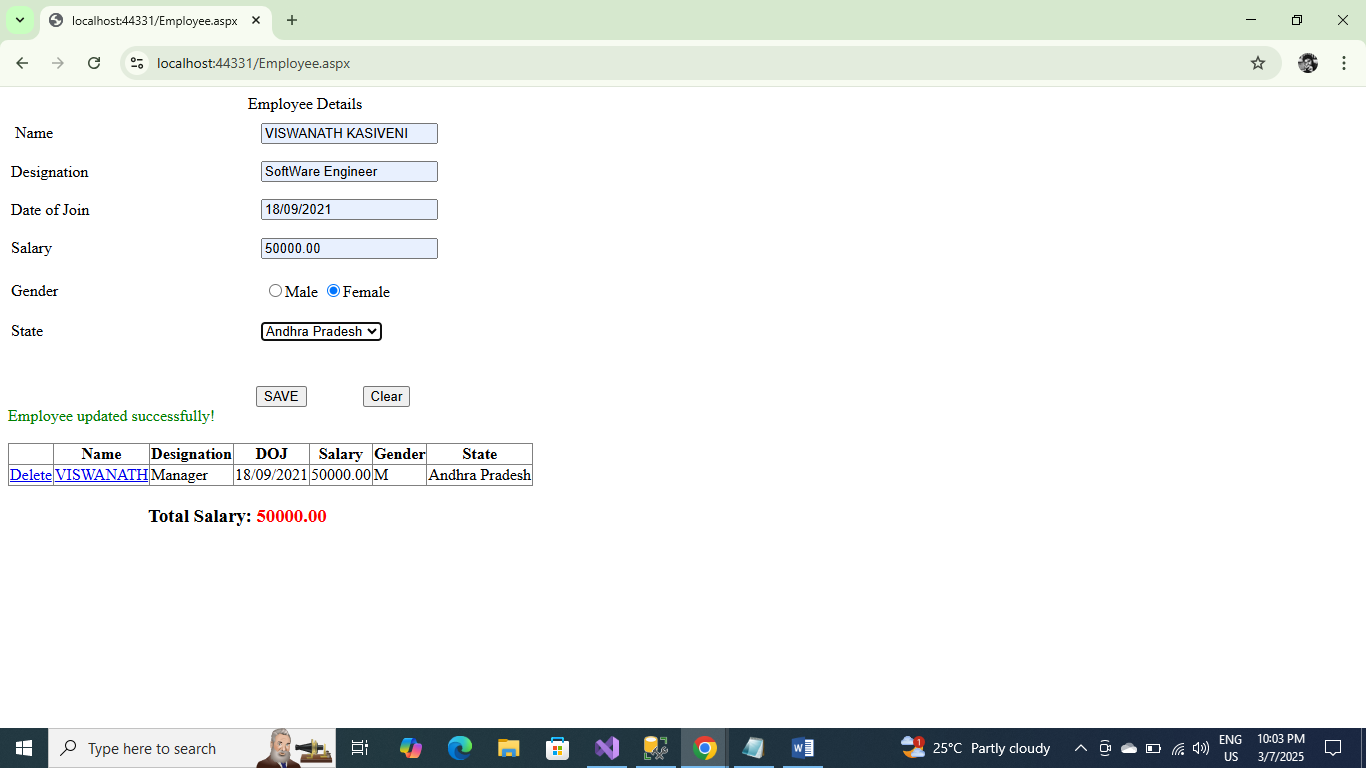
10. Upon updating values, check all validations and data formats as well.



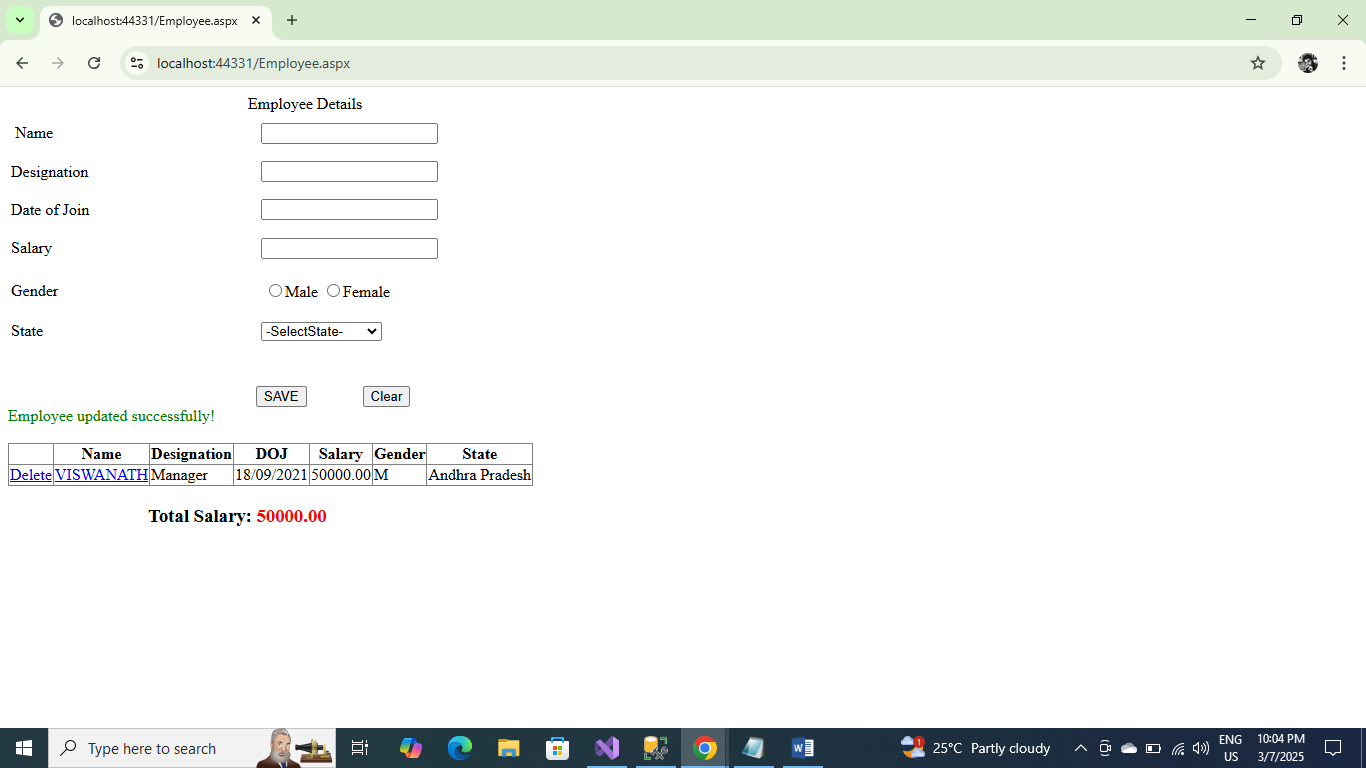
11. Display editable details based on the selected field. When the user clicks the Save button, update the values, show a 'Update Successful' message, and clear all fields.

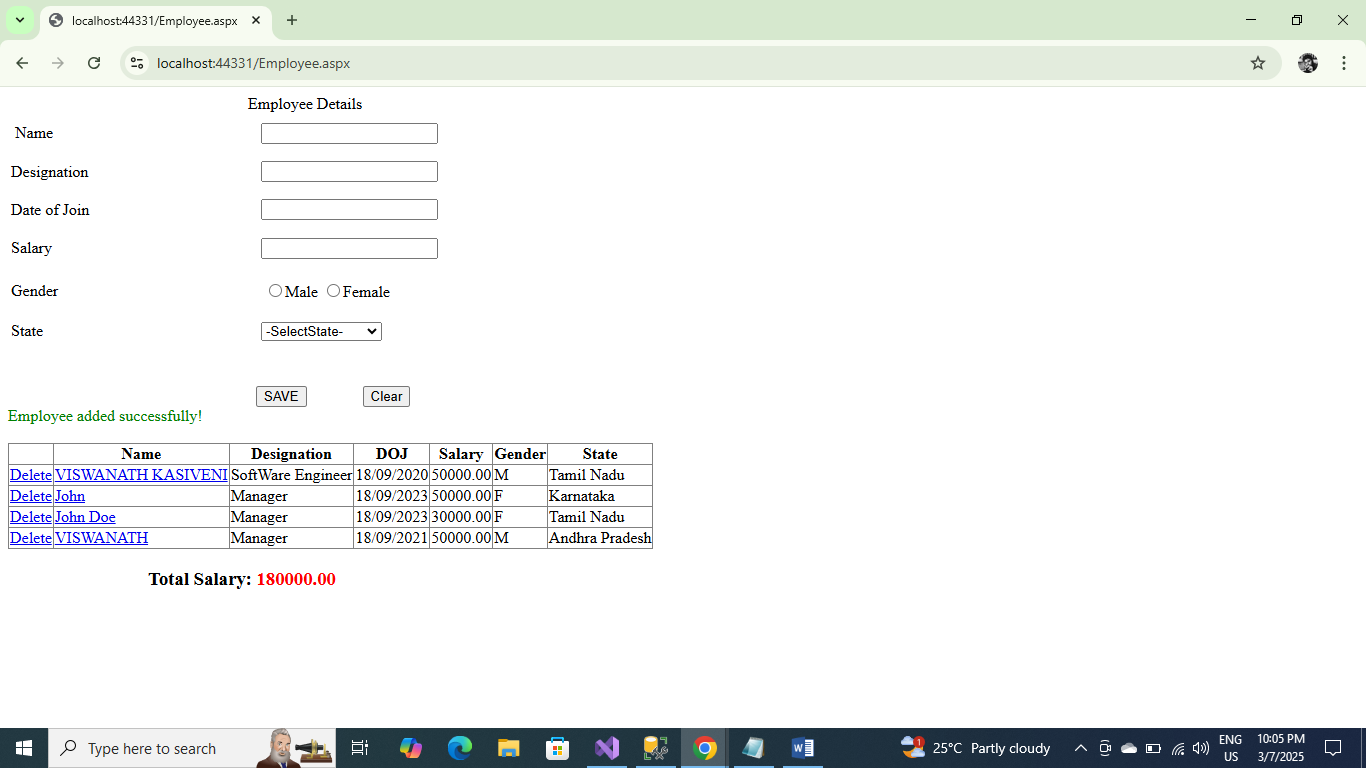


12. The user enters values.

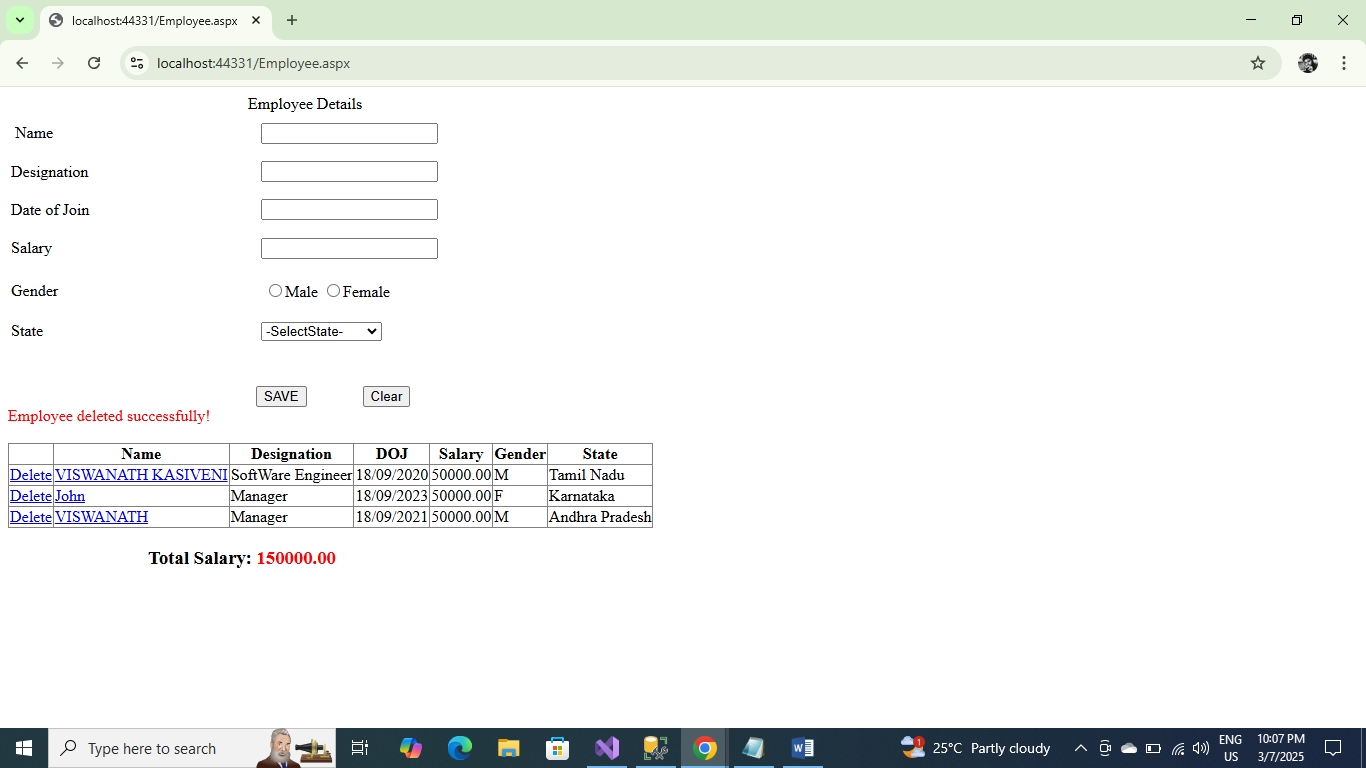


13. We need to clear all values without saving them...



14. While adding an employee, automatically calculate the sum of salaries .

**15.** When the user deletes or updates a record, the sum of salaries will be updated accordingly.



stored procedures:

select \* from Employees

CREATE TABLE Employees (

Id INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100),

Designation NVARCHAR(100),

DateOfJoin DATE,

Salary DECIMAL(10,2),

Gender CHAR(1), -- 'M' or 'F'

State NVARCHAR(100)

);

CREATE PROCEDURE sp\_SaveEmployee

@Id INT = NULL, -- If NULL, INSERT; Otherwise, UPDATE

@Name NVARCHAR(100),

@Designation NVARCHAR(100),

@DateOfJoin DATE,

@Salary DECIMAL(10,2),

@Gender CHAR(1),

@State NVARCHAR(100)

AS

BEGIN

SET NOCOUNT ON;

IF @Id IS NULL

BEGIN

-- Insert New Employee

INSERT INTO Employees (Name, Designation, DateOfJoin, Salary, Gender, State)

VALUES (@Name, @Designation, @DateOfJoin, @Salary, @Gender, @State);

END

ELSE

BEGIN

-- Update Existing Employee

UPDATE Employees

SET Name = @Name,

Designation = @Designation,

DateOfJoin = @DateOfJoin,

Salary = @Salary,

Gender = @Gender,

State = @State

WHERE Id = @Id;

END

END;

CREATE PROCEDURE sp\_DeleteEmployee

@Id INT

AS

BEGIN

SET NOCOUNT ON;

DELETE FROM Employees WHERE Id = @Id;

END;

CREATE PROCEDURE sp\_GetEmployees

AS

BEGIN

SET NOCOUNT ON;

SELECT Id, Name, Designation, DateOfJoin, Salary, Gender, State

FROM Employees

ORDER BY Id DESC;

END;

Reports stored procedures:

CREATE PROCEDURE sp\_Report\_EmployeeWise

@EmployeeId INT

AS

BEGIN

SELECT \* FROM Employees WHERE Id = @EmployeeId;

END

CREATE PROCEDURE sp\_Report\_DesignationWise

@Designation NVARCHAR(100)

AS

BEGIN

SELECT \* FROM Employees WHERE Designation = @Designation;

END

CREATE PROCEDURE sp\_Report\_DesignationCombinationWise

@DesignationList NVARCHAR(MAX) -- Pass comma-separated values

AS

BEGIN

SELECT \* FROM Employees WHERE Designation IN (SELECT value FROM STRING\_SPLIT(@DesignationList, ','));

END

CREATE PROCEDURE sp\_Report\_EmployeeHierarchy

AS

BEGIN

SELECT \* FROM Employees ORDER BY Designation;

END

Project Code..

Employee.cs file code…

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Globalization;

using System.Configuration;

using System.Net.NetworkInformation;

namespace Bookxpert\_Project

{

public partial class Employee : System.Web.UI.Page

{

private readonly string connString = ConfigurationManager.ConnectionStrings["Constr"].ConnectionString;

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

LoadEmployees();

}

private void LoadEmployees()

{

using (SqlConnection conn = new SqlConnection(connString))

{

using (SqlCommand cmd = new SqlCommand("sp\_GetEmployees", conn))

{

cmd.CommandType = CommandType.StoredProcedure;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

gvEmployees.DataSource = dt;

gvEmployees.DataBind();

}

}

}

protected void btnSave\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection conn = new SqlConnection(connString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand("sp\_SaveEmployee", conn))

{

cmd.CommandType = CommandType.StoredProcedure;

// Check if Employee is being Updated or Inserted

int empId = string.IsNullOrEmpty(hdnEmployeeId.Value) ? 0 : Convert.ToInt32(hdnEmployeeId.Value);

cmd.Parameters.AddWithValue("@Id", empId > 0 ? (object)empId : DBNull.Value);

cmd.Parameters.AddWithValue("@Name", txtName.Text.Trim());

cmd.Parameters.AddWithValue("@Designation", txtDesignation.Text.Trim());

// Parse DateOfJoin

DateTime doj;

if (DateTime.TryParseExact(txtDOJ.Text.Trim(), "dd/MM/yyyy", CultureInfo.InvariantCulture, DateTimeStyles.None, out doj))

{

cmd.Parameters.AddWithValue("@DateOfJoin", doj);

}

else

{

lblMessage.ForeColor = System.Drawing.Color.Red;

lblMessage.Text = "Invalid Date Format! Please use dd/MM/yyyy.";

return;

}

cmd.Parameters.AddWithValue("@Salary", Convert.ToDecimal(txtSalary.Text.Trim()));

cmd.Parameters.AddWithValue("@Gender", rblGender.SelectedValue);

cmd.Parameters.AddWithValue("@State", ddlState.SelectedValue);

cmd.ExecuteNonQuery();

// Success Message

lblMessage.ForeColor = System.Drawing.Color.Green;

lblMessage.Text = empId > 0 ? "Employee updated successfully!" : "Employee added successfully!";

// Clear Fields & Reset Hidden Field

ClearFields();

hdnEmployeeId.Value = "";

}

}

// Refresh GridView

LoadEmployees();

}

catch (FormatException ex)

{

lblMessage.ForeColor = System.Drawing.Color.Red;

lblMessage.Text = "Error: Invalid input format. " + ex.Message;

}

}

protected void gvEmployees\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

try

{

// Check if DataKeys are set and retrieve Employee ID

if (gvEmployees.DataKeys[e.RowIndex] != null)

{

int employeeId = Convert.ToInt32(gvEmployees.DataKeys[e.RowIndex].Value);

// Call Delete function

DeleteEmployee(employeeId);

// Reload GridView

LoadEmployees();

lblMessage.ForeColor = System.Drawing.Color.Red;

// Show success message (Optional)

lblMessage.Text = "Employee deleted successfully!";

}

}

catch (Exception ex)

{

lblMessage.Text = "Error: " + ex.Message;

}

}

private void DeleteEmployee(int employeeId)

{

using (SqlConnection con = new SqlConnection(connString))

{

using (SqlCommand cmd = new SqlCommand("sp\_DeleteEmployee", con))

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Id", employeeId);

con.Open();

cmd.ExecuteNonQuery();

}

}

}

protected void ClearFields()

{

txtName.Text = "";

txtDesignation.Text = "";

txtDOJ.Text = "";

txtSalary.Text = "";

rblGender.ClearSelection();

ddlState.SelectedIndex = 0;

}

protected void gvEmployees\_RowCommand(object sender, GridViewCommandEventArgs e)

{

if (e.CommandName == "EditRecord")

{

int empId = Convert.ToInt32(e.CommandArgument);

using (SqlConnection conn = new SqlConnection(connString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand("SELECT \* FROM Employees WHERE Id = @Id", conn))

{

cmd.Parameters.AddWithValue("@Id", empId);

using (SqlDataReader reader = cmd.ExecuteReader())

{

if (reader.Read())

{

hdnEmployeeId.Value = reader["Id"].ToString();

txtName.Text = reader["Name"].ToString();

txtDesignation.Text = reader["Designation"].ToString();

txtDOJ.Text = Convert.ToDateTime(reader["DateOfJoin"]).ToString("dd/MM/yyyy");

txtSalary.Text = reader["Salary"].ToString();

rblGender.SelectedValue = reader["Gender"].ToString();

ddlState.SelectedValue = reader["State"].ToString();

}

}

}

}

}

}

private DataTable GetEmployeeById(int id)

{

DataTable dt = new DataTable();

using (SqlConnection conn = new SqlConnection(connString))

{

string query = "SELECT Id, Name, Designation, DateOfJoin, Salary, Gender, State FROM Employees WHERE Id = @Id";

using (SqlCommand cmd = new SqlCommand(query, conn))

{

cmd.Parameters.AddWithValue("@Id", id);

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

conn.Open();

da.Fill(dt);

}

}

}

return dt; // Now it will return actual data from DB

}

protected void Btn\_Click(object sender, EventArgs e)

{

ClearFields();

}

}

}

Employee.ASPX file code..

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Employee.aspx.cs" Inherits="Bookxpert\_Project.Employee" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.5/jquery.validate.min.js"></script>

<title></title>

<style type="text/css">

.auto-style1 {

width: 100%;

height: 237px;

}

.auto-style2 {

width: 246px;

}

.error {

color: red;

font-size: 14px;

font-weight: bold;

}

</style>

<script>

$(document).ready(function () {

var total = 0;

// Loop through each salary cell inside the GridView

$(".salary").each(function () {

total += parseFloat($(this).text()) || 0;

});

// Update the ASP.NET Label with Total Salary

$("#<%= lblTotalSalary.ClientID %>").text(total.toFixed(2));

});

</script>

</head>

<body>

<script>

$(document).ready(function () {

$("#form1").validate({

rules: {

txtName: {

required: true,

minlength: 3

},

txtDesignation: {

required: true

},

txtSalary: {

required: true,

number: true

},

rblGender: {

required: true

},

ddlState: {

required: true

}

},

messages: {

txtName: {

required: "Please enter your name",

minlength: "Name must be at least 3 characters long"

},

txtDesignation: {

required: "Please enter designation"

},

txtSalary: {

required: "Please enter salary",

number: "Only numbers are allowed"

},

rblGender: {

required: "Please select gender"

},

ddlState: {

required: "Please select a state"

}

},

errorPlacement: function (error, element) {

if (element.attr("type") === "radio") {

error.insertAfter(element.closest("table"));

} else {

error.insertAfter(element);

}

}

});

// Custom validation for DropDownList

$.validator.addMethod("valueNotEmpty", function (value, element) {

return value !== ""; // Ensures a state is selected

}, "Please select a state");

$("#ddlState").rules("add", { valueNotEmpty: true });

});

</script>

<form id="form1" runat="server">

<div>

Employee Details</div>

<table class="auto-style1">

<tr>

<td class="auto-style2"> Name </td>

<td><asp:TextBox ID="txtName" runat="server" ClientIDMode="Static"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Designation</td>

<td><asp:TextBox ID="txtDesignation" runat="server" ClientIDMode="Static"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Date of Join </td>

<td><asp:TextBox ID="txtDOJ" runat="server" ClientIDMode="Static"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Salary</td>

<td><asp:TextBox ID="txtSalary" runat="server" ClientIDMode="Static"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Gender</td>

<td><asp:RadioButtonList ID="rblGender" runat="server" RepeatDirection="Horizontal" ClientIDMode="Static">

<asp:ListItem Text="Male" Value="M"></asp:ListItem>

<asp:ListItem Text="Female" Value="F"></asp:ListItem>

</asp:RadioButtonList></td>

</tr>

<tr>

<td class="auto-style2">State</td>

<td> <asp:DropDownList ID="ddlState" runat="server" ClientIDMode="Static">

<asp:ListItem Text="-SelectState-" Value=""></asp:ListItem>

<asp:ListItem Text="Andhra Pradesh" Value="Andhra Pradesh"></asp:ListItem>

<asp:ListItem Text="Karnataka" Value="Karnataka"></asp:ListItem>

<asp:ListItem Text="Tamil Nadu" Value="Tamil Nadu"></asp:ListItem>

</asp:DropDownList></td>

</tr>

</table>

<br />

<br />

<asp:Button ID="btnSave" runat="server" Text="SAVE" OnClick="btnSave\_Click" />

<asp:Button ID="Btn" runat="server" Text="Clear" OnClick="Btn\_Click" />

<br />

<asp:Label ID="lblMessage" runat="server"></asp:Label>

<br />

<br />

<asp:HiddenField ID="hdnEmployeeId" runat="server" />

<asp:GridView ID="gvEmployees" runat="server" AutoGenerateColumns="False" DataKeyNames="Id"

PageSize="2"

AutoGenerateDeleteButton="True" OnRowDeleting="gvEmployees\_RowDeleting" OnRowCommand="gvEmployees\_RowCommand">

<Columns>

<asp:TemplateField HeaderText="Name">

<ItemTemplate>

<asp:LinkButton ID="lnkEdit" runat="server" Text='<%# Eval("Name") %>'

CommandName="EditRecord" CommandArgument='<%# Eval("Id") %>' />

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Designation" HeaderText="Designation" />

<asp:BoundField DataField="DateOfJoin" HeaderText="DOJ" DataFormatString="{0:dd/MM/yyyy}" />

<asp:TemplateField HeaderText="Salary">

<ItemTemplate>

<span class="salary"><%# Eval("Salary") %></span>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Gender" HeaderText="Gender" />

<asp:BoundField DataField="State" HeaderText="State" />

</Columns>

</asp:GridView>

<h3> Total Salary: <asp:Label ID="lblTotalSalary" runat="server" ForeColor="Red"></asp:Label></h3>

</form>

</body>

</html>

WEB.Config file code…

<?xml version="1.0" encoding="utf-8"?>

<!--

For more information on how to configure your ASP.NET application, please visit

https://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<connectionStrings>

<add name="Constr"

connectionString="Data Source=DESKTOP-OM44QHM\SQLSERVER;Initial Catalog=Viswa;Integrated Security=True"

providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.7.2" />

<httpRuntime targetFramework="4.7.2" />

</system.web>

<system.codedom>

<compilers>

<compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:1659;1699;1701" />

<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:41008 /define:\_MYTYPE=\"Web\" /optionInfer+" />

</compilers>

</system.codedom>

</configuration>