



L OVELY
P ROFESSIONAL
U NIVERSITY

PROJECT REPORT
ON
PAYROLL MANAGEMENT SYSTEM

Modern Web Technology

Submitted by: -

S.V.Sai Krishna(11605879-14)

K V Durga Prasad(11606853-48)

C S Sanjeev Bhargav(11607883-52)

S.Tharun Raj(11612167-60)

Submitted to:-

Ankita Wadhawan(23891)

SYNOPSIS

“Payroll Management” is a distributed application, developed to evaluate the performance of employees working in any organization. It maintains the information about a company, personal details of their employees, also the project details assigned to particular developer. The application is actually a suite of applications developed using Java.

It is simple to understand and can be used by anyone who is not even familiar with simple employees’ system. It is user friendly and just asks the user to follow step by step operations by giving him few options. It is fast and can perform many operations of a company.

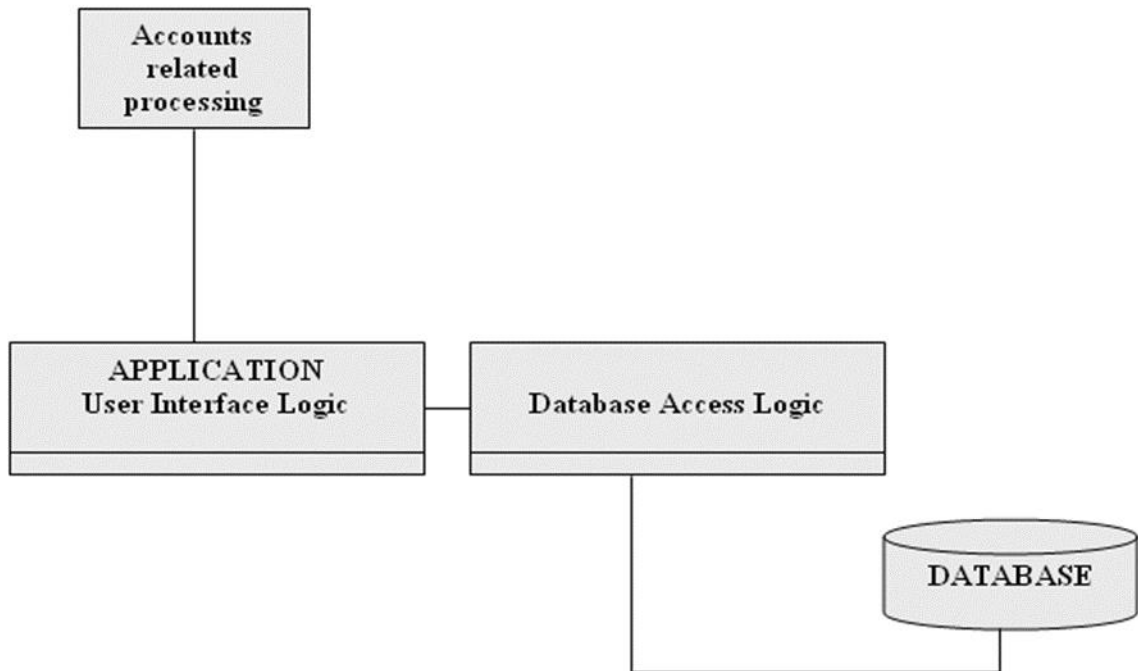
This software package has been developed using the powerful coding tools of JAVA at Front End and Microsoft Access at Back End. Because of the Visual features, the software is very user friendly. The package contains different modules like Contacts, Search for property and other useful Links. This version of the software has multi-user approach. For further enhancement or development of the package, user’s feedback will be considered.

This project basically deals with five modules and their further sub modules. First module is the employee module into which we can enter employee details such as his name, address, phone number, his basic salary and many more. After that we can view the details further by using the employee id, and we can modify the details also. Similarly, in department we have the details of all the HOD’s of the departments. Next comes the salary module in this we can view the salary issued to the employee. And can issue them to the employee we want to and can fix it to them.

We can put the grades also in the grade module. Basically, we create the grades in this with specified details. We can view the details accordingly. We can just enter the grade name and can view the details encapsulated in the grade. Other than this we can view the whole thing all together by getting into the view grade section.

We can view the report also in this. It can be viewed in this easily that to which the monthly salary has been allotted and for which month. We can also look out those employees to whom the salary has not been issued and the further details also such as for which month it has not been issued. This is the basic overview of the whole project

DATABASE DESIGN



Database

We have various tables in our project namely:

- Employee Table
- Department table
- Grade Table
- Salary Table

All the above tables are now briefly explained in which the Primary key and the Data Type of all fields are discussed.

Employee Table

In this Table, we have the various fields to be filled about the employee who are working in the firm. All the personal details of all employees are filled. These fields include Name, ID, EmailID, etc. Of all the employees.

Field Name	Key	Data Type
ID	Primary	Char
Name	-	Char
Age	-	int
Address	-	Int
Contact No.	-	Int
EmailID	-	Char
Department	-	Char
D.O.J	-	Int
Grade	-	Char

Department Table

In this Table, entries for the department name and its HOD name are filled that is the HOD is assigned to the Department. All Departments have their respective HOD.

Field Name	Key	Data Type
Name	Primary	Char
HOD Name	-	Char

Grade Table

Grade Table takes the entries of Basic Salary, HRA, DA, TA, PF, IT and Net Salary are being filed and for the particular Grade, all these values are assigned. Grade is assigned according to the job of the employee.

Field Name	Key	Data Type
Name	Primary	Char
HRA	-	Int
TA	-	Int
DA	-	Int
PF	-	Int
IT	-	Int

Salary Table

In Salary Table also, we have the fields for Basic Salary, D.O..J, HRA, DA, TA, PF, IT. For the particular salary, the values of all these fields are assigned.

Field Name	Key	Data Type
ID	Primary	Char
BS	-	Int
DOJ	-	Int
HRA	-	Int

DA	-	Int
TA	-	Int
PF	-	Int
IT	-	Int
Net Salary	-	Int
Month	-	Char

Basically, in our project have the main modules regarding the project that which appears the front page of the project. This is the main menu of our project. This is the front page or the first page which appears when we start our project

It has 5 modules namely:

- Employee
- Department
- Salary
- Grade
- Report

➤ Employee: -

It is the first sub-module of employee module.in this form, the personal details of each employee Is filled and saved. Personal details are like emp id name, last name, address, date of joining, email id, department name, basic salary of the employee and many more.

➤ Department: -

The new HOD is assigned to the respective departments and then the record is saved in the database by pressing the save button. The new department frame can also be canceled using the cancel button.

➤ Salary: -

The salary of all the employees can be viewed including their HRA, TA, DA, PF, etc. in this table we can view all the salary related details of the employee with respect to his employee id.

➤ Report: -

All the information about an employee can be viewed in this section. All the pre decided things come here first and then we basically enter the date and the respected month and the year. And then press the issue button.

SYSTEM DESIGN

Design methodology: -

A software require specification document tells us what a system does and becomes input to the design process. The purpose of design phase is to produce a solution to problem given SRS document.

TABLES: -

Mainly, in this project we have four tables that are employee table, department table, salary table and grade table. After filling all the entries, the detail of all the employees are ready.

ADVANTAGES: 1. Cost effective

2. Simple to operate

Design Process

The computer system design process is an exercise of specifying how, the system will work. It is an iterative process, which is based on what the system will be do as shown in the feasibility report. Mainly, following five parts have been included in the system design process

Output Design

The starting point of the design process is the proper knowledge of system requirements which will normally be converted in terms of output.

Input Design

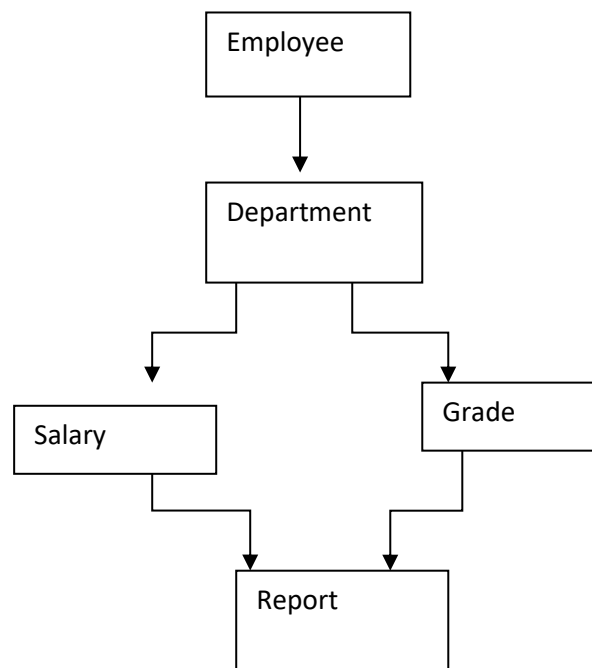
Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or new documents may have to be introduced.

File Design

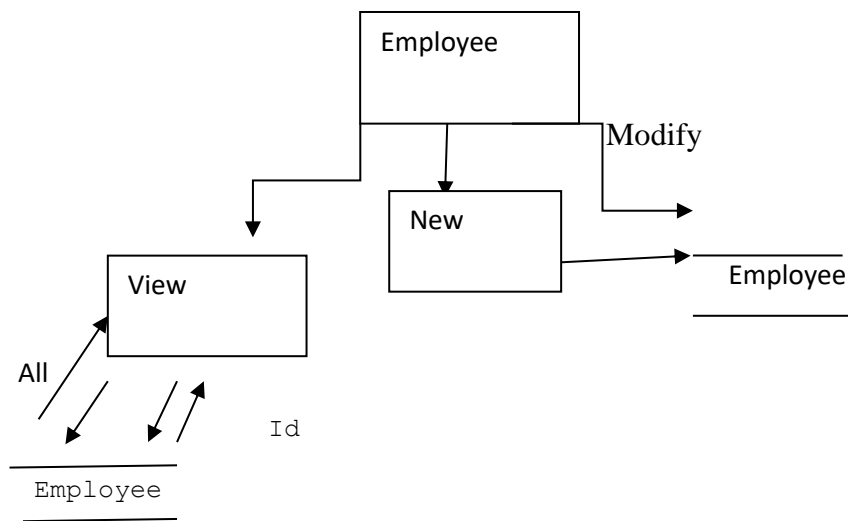
Once the input data is captured in the system, these may to be preserved either for a short or long period. These data will generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files.

DFD

0 LEVEL DFD



1st LEVEL DFD:



IMPLEMENTATION CODE

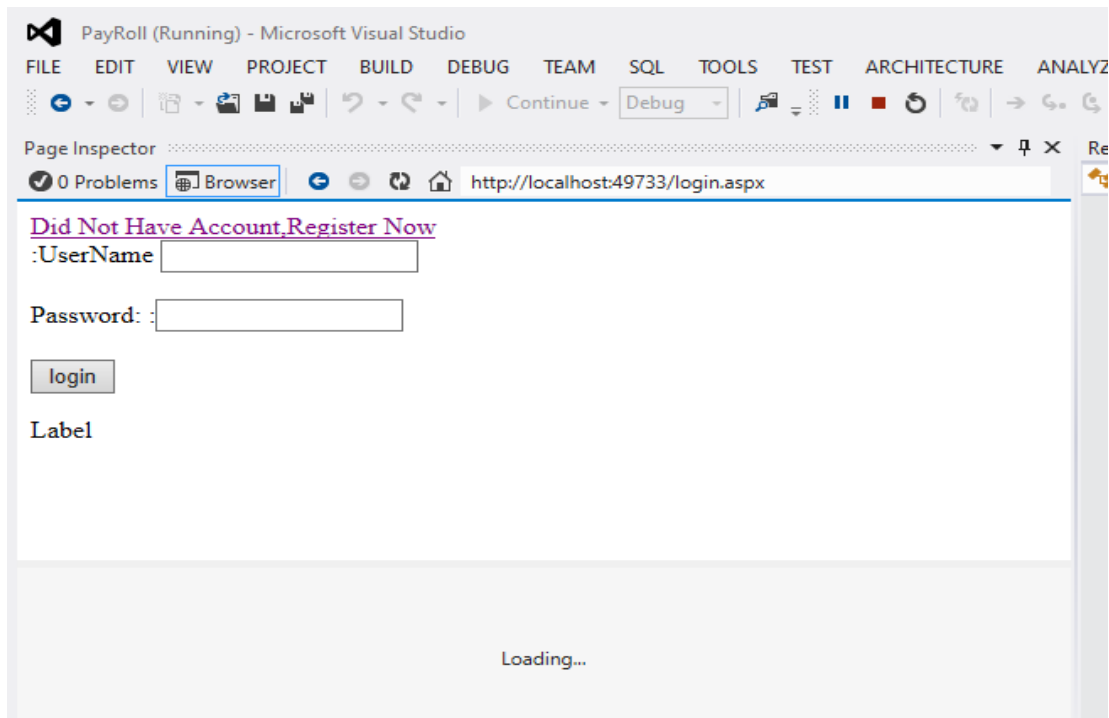
Login Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

namespace PayRoll
{
    public partial class login : System.Web.UI.Page
    {
        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\PayRoll.mdf;Integrated
Security=True");
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            String check = "select count(*) from [Register] where Email
Id='"+UserTxt.Text+"' and Password='"+passtxt.Text+"'";
            SqlCommand cmd = new SqlCommand(check, con);
            con.Open();
            int temp = convert.ToString(cmd. ExecuteScalar().ToString()) ;
            con.Close();
            if (temp == 1)
            {
                Response.Redirect("~/DashBoard.aspx");
            }
            else
            {
                Label1.ForeColor=System.Drawing.Color.Red;
                Label1.Text = "Your Email Id or Password is Invalid";
            }
        }
    }
}
```

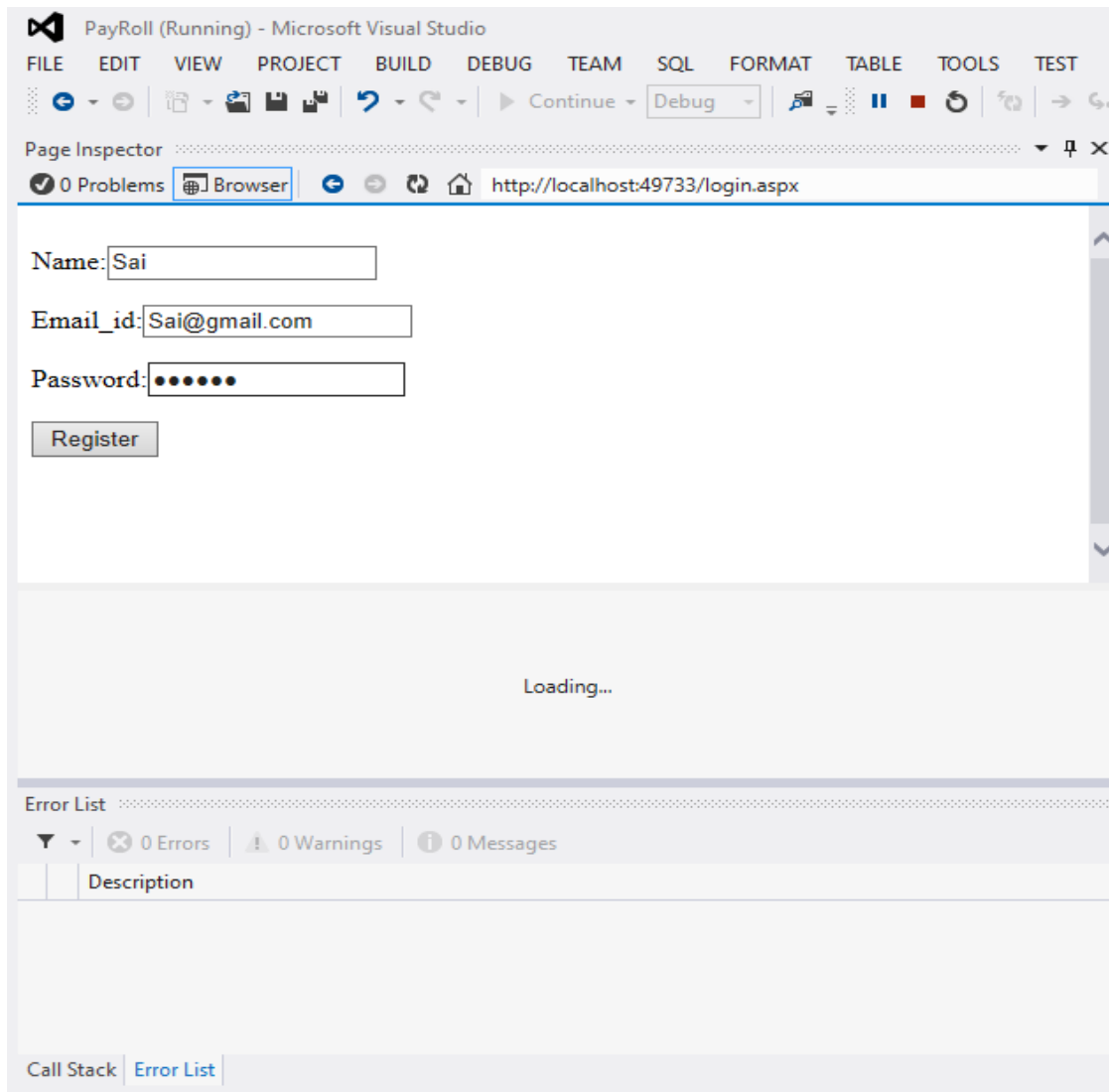


➤ Register Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace PayRoll
{
    public partial class Register : System.Web.UI.Page
    {
        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\PayRoll.mdf;Integrated
Security=True");
        protected void Page_Load(object sender, EventArgs e)
        {

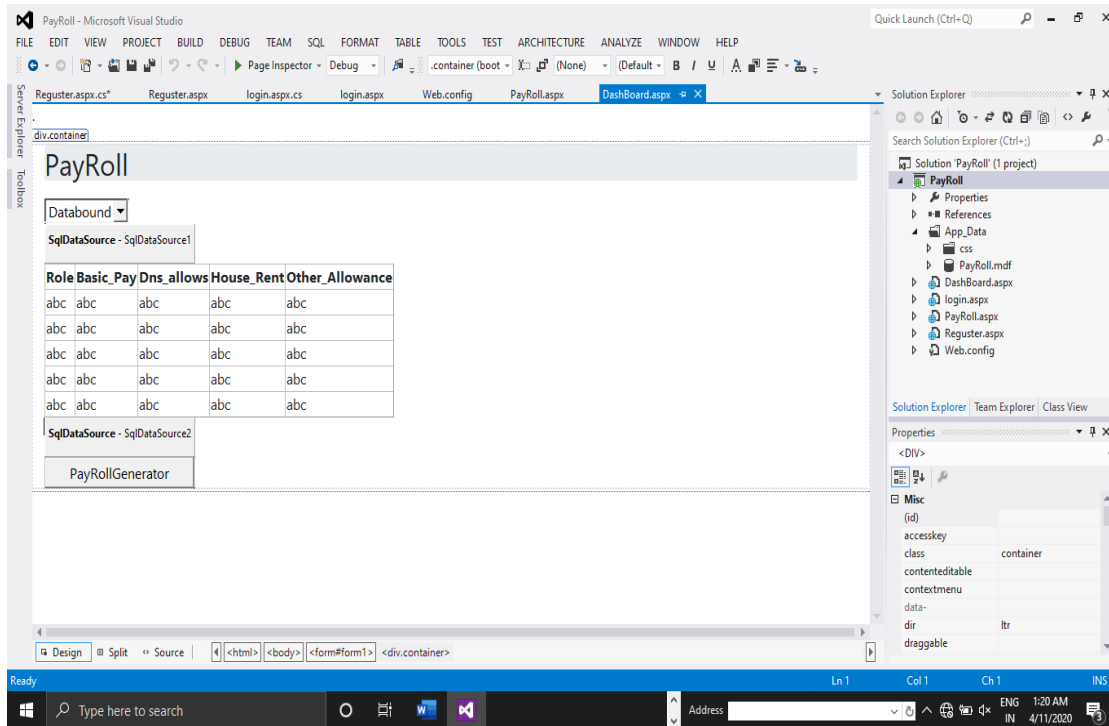
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            String Ins="Insert into [regiter] (Name,Email Id,Password)
values('"+nametxt.Text+"','"+Emailtxt.Text+"','"+Passtxt.Text+"')";
            SqlCommand cmd = new SqlCommand(Ins, con);
            con.Open();
            cmd.BeginExecuteReader();
            con.Close();
            Response.Redirect("~/login.aspx");
        }
    }
}
```



➤ **Dashboard Page**

Design Page for Dashboard



Source Code for Dashboards

```
<head runat="server">
    <title>DashBoard </title>
    <link href="App_Data/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <form id="form1" runat="server">
        <div class="container">
            <div class="breadcrumb">
                <h1>PayRoll</h1>
            </div>
            <asp:DropDownList ID="emp" runat="server" DataSourceID="SqlDataSource1"
DataTextField="EmpNmae" DataValueField="EmpRole"
AutoPostBack="true"></asp:DropDownList>
            <asp:SqlDataSource runat="server" ID="SqlDataSource1"
ConnectionString='<%$ ConnectionStrings:ConnectionString %>' SelectCommand="SELECT *
FROM [Employee] ORDER BY [EmpNmae]"></asp:SqlDataSource>

            <asp:GridView runat="server" AutoGenerateColumns="False"
DataKeyNames="Role" DataSourceID="SqlDataSource2">

                <Columns>
                    <asp:BoundField DataField="Role" HeaderText="Role" ReadOnly="True"
SortExpression="Role"></asp:BoundField>
                    <asp:BoundField DataField="Basic_Pay" HeaderText="Basic_Pay"
SortExpression="Basic_Pay"></asp:BoundField>
                    <asp:BoundField DataField="Dns_allows" HeaderText="Dns_allows"
SortExpression="Dns_allows"></asp:BoundField>
                    <asp:BoundField DataField="House_Rent" HeaderText="House_Rent"
SortExpression="House_Rent"></asp:BoundField>
                    <asp:BoundField DataField="Other_Allowance"
HeaderText="Other_Allowance" SortExpression="Other_Allowance"></asp:BoundField>
                </Columns>
            </asp:GridView>
        </div>
    </form>
    <div class="container">
        <div class="text-center">
            <asp:SqlDataSource ID="SqlDataSource2" runat="server"
                ConnectionString="Data Source=PayRoll.mdf; AttachDbFilename=PayRoll.mdf;
                User ID=sa; Password=; MultiplePageTimeOut=30000;"
                SelectCommand="SELECT * FROM [Employee] ORDER BY [EmpNmae]">
            </asp:SqlDataSource>
            <asp:GridView ID="GridView1" runat="server" DataSourceID="SqlDataSource2">
                <Columns>
                    <asp:BoundField DataField="Role" HeaderText="Role" ReadOnly="True"
                        SortExpression="Role">
                    </asp:BoundField>
                    <asp:BoundField DataField="Basic_Pay" HeaderText="Basic_Pay"
                        SortExpression="Basic_Pay">
                    </asp:BoundField>
                    <asp:BoundField DataField="Dns_allows" HeaderText="Dns_allows"
                        SortExpression="Dns_allows">
                    </asp:BoundField>
                    <asp:BoundField DataField="House_Rent" HeaderText="House_Rent"
                        SortExpression="House_Rent">
                    </asp:BoundField>
                    <asp:BoundField DataField="Other_Allowance" HeaderText="Other_Allowance"
                        SortExpression="Other_Allowance">
                    </asp:BoundField>
                </Columns>
            </asp:GridView>
        </div>
    </div>
    <div class="text-center">
        <asp:Button ID="PayRollGenerator" runat="server" Text="PayRollGenerator" />
    </div>
</body>
</html>
```

```

        <asp:SqlDataSource runat="server" ID="SqlDataSource2"
ConnectionString='<%$ ConnectionStrings:ConnectionString %>' SelectCommand="SELECT *
FROM [Salary] WHERE ([Role] = @Role)">
        <SelectParameters>
            <asp:ControlParameter ControlID="emp" PropertyName="SelectedValue"
Name="Role" Type="String"></asp:ControlParameter>
        </SelectParameters>
        </asp:SqlDataSource>
        <asp:Button Text="PayRollGenerator" runat="server"
OnClick="Unnamed2_Click" />

    </div>

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

```

Output for Dashboard

The screenshot shows the PayRoll application running in Microsoft Visual Studio. The browser window displays the following content:

PayRoll

Bhargav ▾

Role	Basic_Pay	Dns_allows	House_Rent	Other_Allowance
Sales Manger	25000	12	50	16

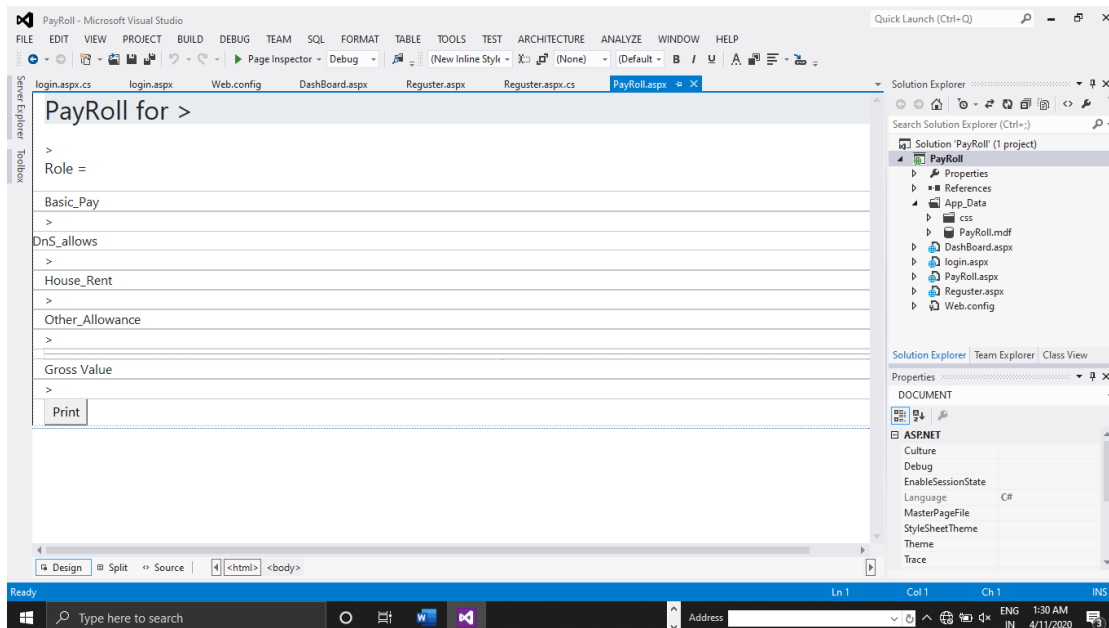
PayRollGenerator

Loading...

The Error List at the bottom shows 0 Errors, 0 Warnings, and 0 Messages.

➤ Payroll Generator Page

Design Page for Payroll Generator Page



Source code for Payroll Generator Page

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

    <div class="container">
        <div class="breadcrumb">
            <h1>PayRoll for <%=Session["name"] %></h1>
        </div>
        <div class="text-justify">
            <label><%=DateTime.Today %></label><br />
            <h3 class="text-justify">Role =<b><%=Session["role"] %></b></h3>
            <div class="row">
                <div class="col-lg-6">
                    Basic_Pay

                </div>
                <div class="col-lg-6">
                    <%=Session["basic"] %>
                </div>
            </div>

            <div class="row">
                <div class="auto-style1">
                    DnS_allows
                </div>
                <div class="col-lg-6">
                    <%=Session["Dns"] %>
                </div>
            </div>

            <div class="row">
                <div class="col-lg-6">
                    House_Rent
```

```

        </div>
        <div class="col-lg-6">
            <%=Session["House_Rent"] %>
        </div>
    </div>

    <div class="row">
        <div class="col-lg-6">
            Other_Allowance
        </div>
        <div class="col-lg-6">
            <%=Session["Other_Allowance"] %>
        </div>
    </div>
    <hr />
    <div class="row">
        <div class="col-lg-6">
            Gross Value
        </div>
        <div class="col">
            <%=double.Parse(Session["basic"].ToString())+double.Parse(Session["Dns"].ToString())+double.Parse(Session["House_Rent"].ToString())+double.Parse(Session["Other_Allowance"].ToString()) %>
        </div>
    </div>

    <button title="Print" onclick="myprint()">Print</button>
    <script type="text/javascript">
        function myprint() {
            window.print();
        }
    </script>
</div>

</div>

```

Calculations for Employee Salary

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

namespace PayRoll
{
    public partial class DashBoard : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Unnamed2_Click(object sender, EventArgs e)
        {
            String name = emp.SelectedItem.Text;
            String role=emp.SelectedValue;

```

```

        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\PayRoll.mdf;Integrated
Security=True");
        con.Open();
        SqlCommand cmd=new SqlCommand("SELECT * FROM Salary WHERE role='"+ role
+ "'", con);
        SqlDataReader result=cmd.ExecuteReader();
        result.Read();
        double basic = double.Parse(result["Basic_Pay"].ToString());
        double Dns = double.Parse(result["DNS_Allows"].ToString());
        double House_Rent = double.Parse(result["House_Rent"].ToString());
        double Other_Allowance =
double.Parse(result["Other_Allowance"].ToString());

        double Dn_c = basic * 0.12;
        double hr_c = basic * (House_Rent/ 100);
        double ot_c = basic * (Other_Allowance / 100);

        Session["basic"] = basic;
        Session["Dns"] = Dn_c;
        Session["House_Rent"] = hr_c;
        Session["Other_Allowance"] = ot_c;

        Session["name"] = name;
        Session["role"] = role;
        Response.Redirect("PayRoll.aspx");
    }
}
}

```

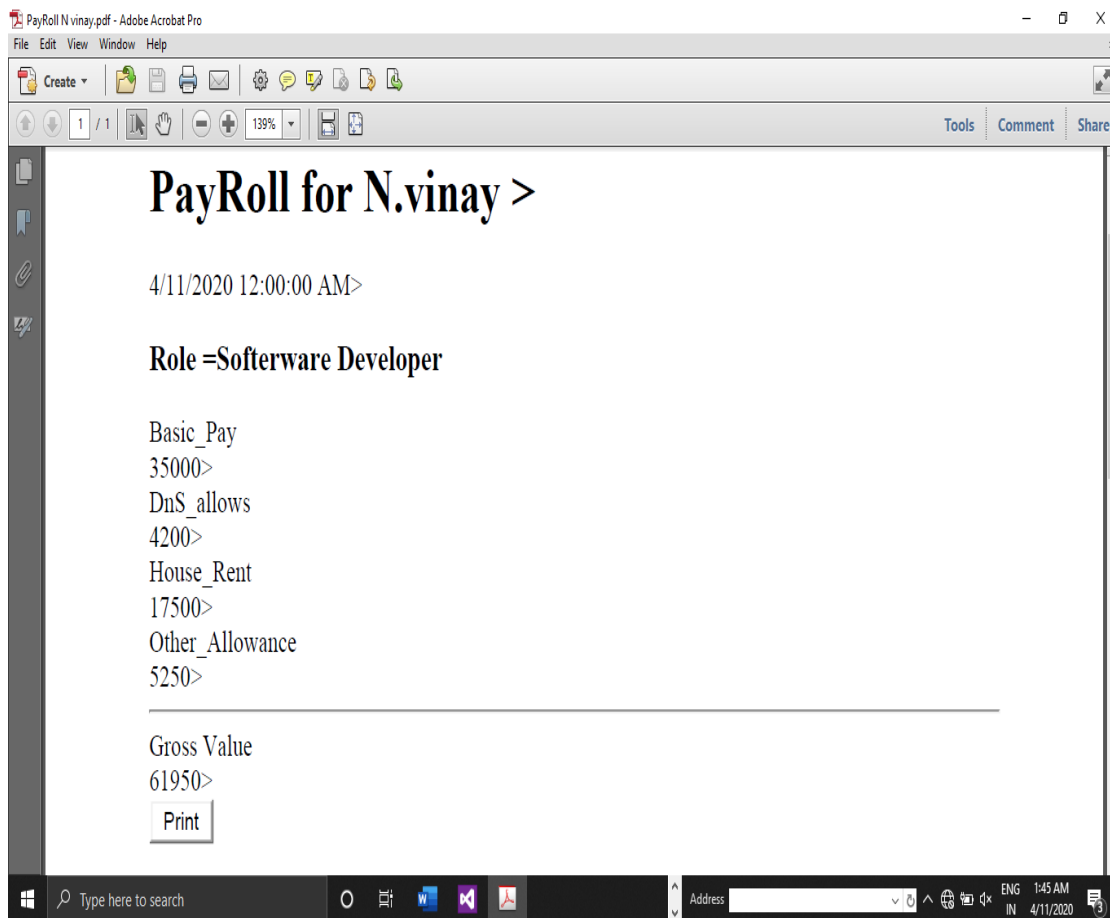
Output for Generator Payroll

The screenshot shows the Visual Studio IDE with the PayRoll.aspx page open in the browser. The page title is "PayRoll for N.vinay >". The content shows the following details:

- Date: 4/11/2020 12:00:00 AM
- Role: Software Developer
- Basic_Pay: 35000
- DnS_allows: 4200
- House_Rent: 17500
- Other_Allowance: 5250
- Gross Value: 61950

The bottom of the screenshot shows the Error List panel, which is currently empty, indicating no errors or warnings.

Output in PDF format



CONCLUSION

- With the theoretical inclination of our syllabus it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The construction of this Minor Project “PAYROLL MANAGEMENT SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer.

The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project

The project also provided us the opportunity of interacting with our teachers and to gain from their vast experience.

- GitHub Links: -

1. <https://github.com/bhargavgit99>
2. <https://github.com/svsaikrishna/INT402.git>

