HR Management System.

Group :

1. Milinda Kaushika – LMU ID - 17030187
2. Mohamed Arham Khan – LMU ID - 17030200
3. K. Aroonpragash – LMU ID - 17030206

Table of Contents

Acknowledgement

First and foremost, we have to thank Ms. Mahesha. This assignment cannot be completed if her dedicated teaching and assistance was not given to us. Her teaching style and enthusiasm for the topic made a strong impression on us. She raised many precious points in our discussion and we hope that we managed to address several of them in this assignment.

Preface

This documentation is for “Knowledge Wisdom”, a well-established educational provider in Sri Lanka. This second course work is focused on the HR Department and providing various information about the courses conducted to its customers.

The main functionalities are:

- Proper attendance handling
- Leave apply and management
- Payroll calculations
- Provide information on courses conducted
- Mange a forum to comment about the courses

The application was developed with Visual Studio C#, ASP.Net and MS SQL Server.

This documentation contains:

- Detail instructions to run the program
- Details of the functionalities built
- Software architecture

Hardware and Software Specification:

HARDWARE:

- Minimum 5 GB HDD space2
- 64 based processor
- Printer (any)
- Power Supply For Backup

SOFTWARE:

- Visual Studio 2014 (ASP.Net)
- .net framework 4.0
- MSSQL server management studio 2014

Visual ASP.Net is great! It's an easy, economical and fast application development tool; it's a good prototyping tool and developer's love using it. Like any high-level programming language, Visual ASP.Net lets the programmer write really awful programs, and with ASP.Net, you can screw up more easily and faster than ever! Important business logic can be attached to GUI widgets rather than placed in reusable objects, making it hard to share and reuse code.

Microsoft SQL Server:

Microsoft SQL Server may be a relational database management system, or DBMS, that supports a large kind of dealing process, business intelligence and analytic applications in IT and business consulting company. It's one among the 3 market-leading database technologies, alongside Oracle database and IBM's DB2.

B) Microsoft SQL Server as the back-end:

Microsoft SQL is a database package generally, used to design database applications. Microsoft SQL is used in this project for following reasons**:**

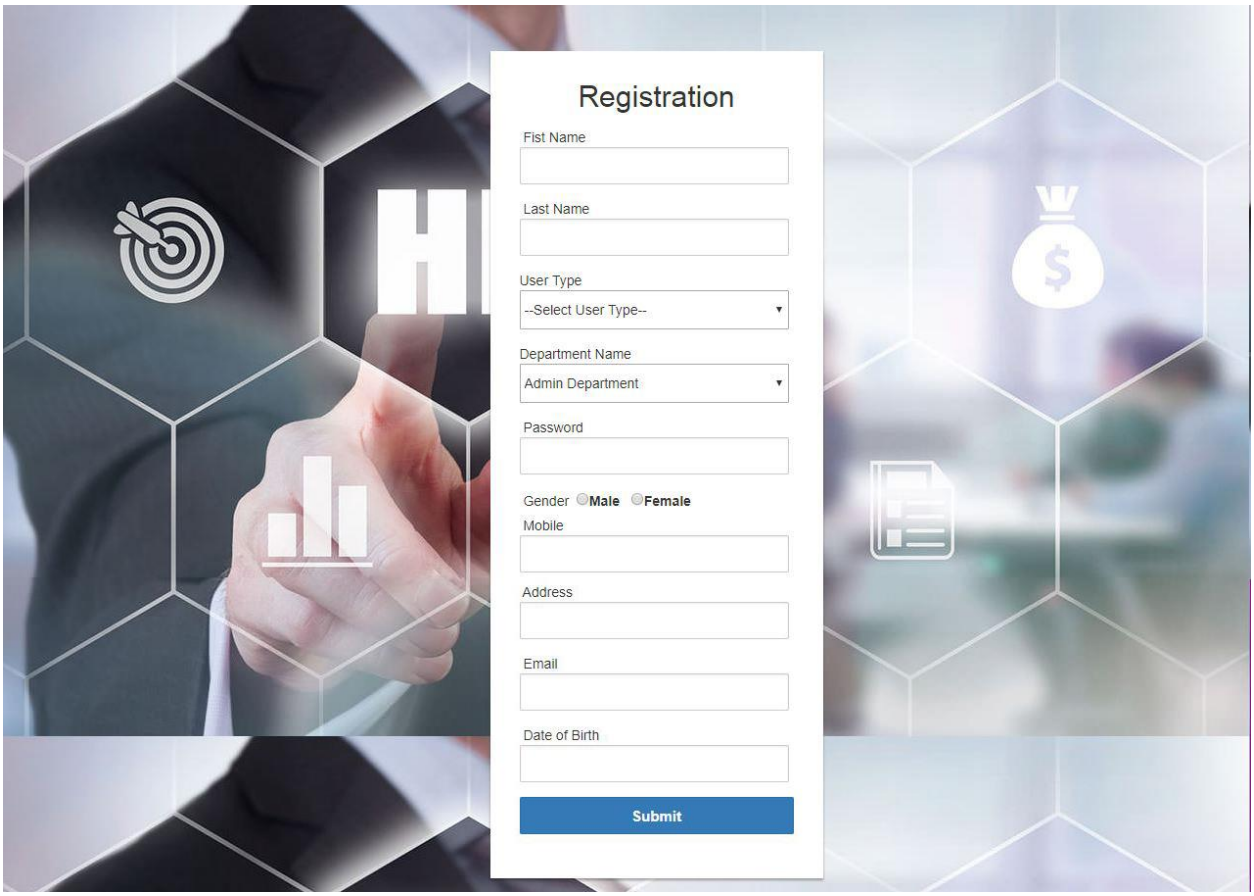
- Microsoft SQL able to store large data.
- It's DBMS.
- Creating relationship is not a complex task.
- Microsoft SQL can execute any valid SQL query.
- It provides all necessary forms of data types.
- Microsoft SQL has good connectivity with visual Asp.Net.

User Interfaces

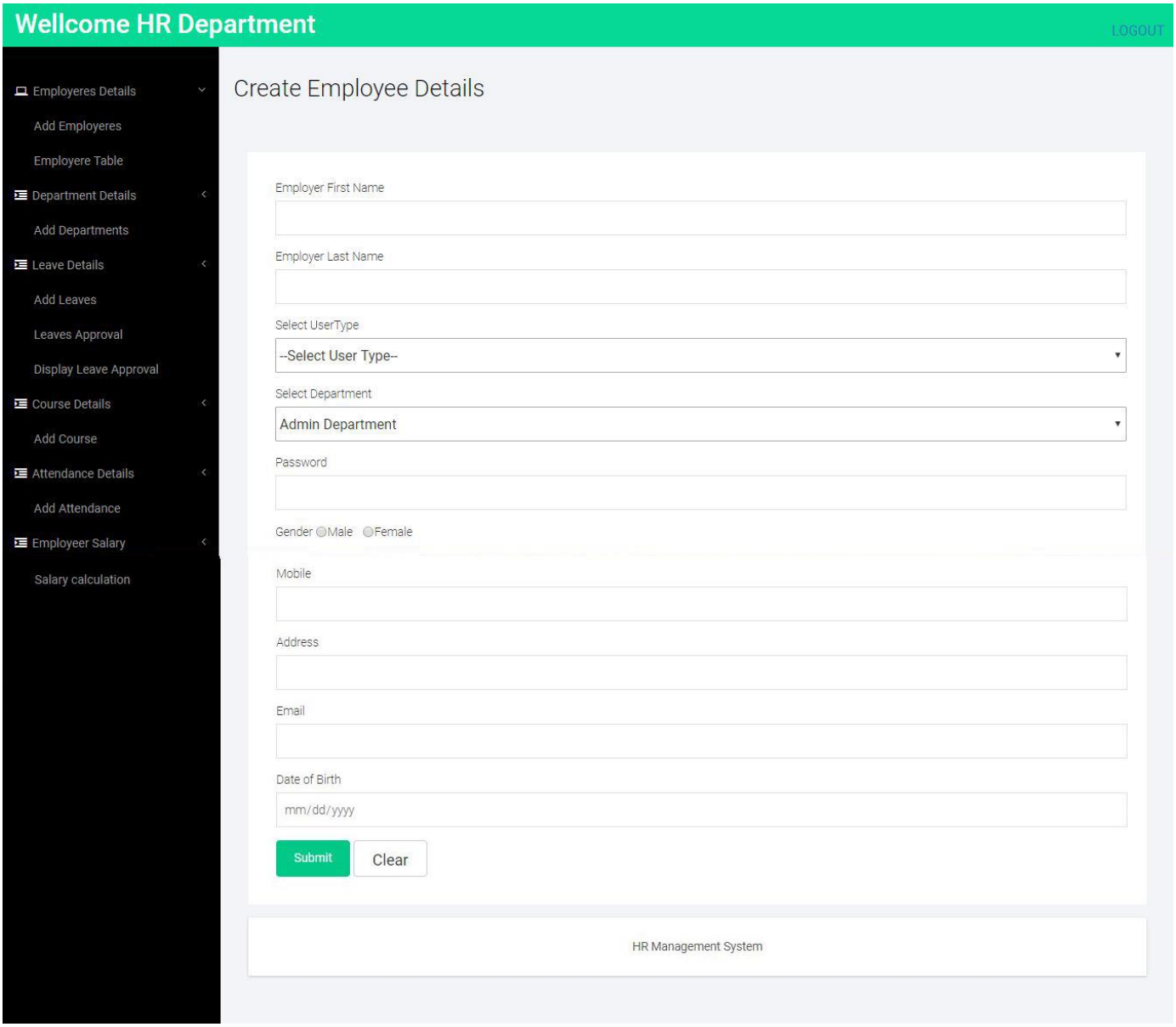
User Management

User Types:

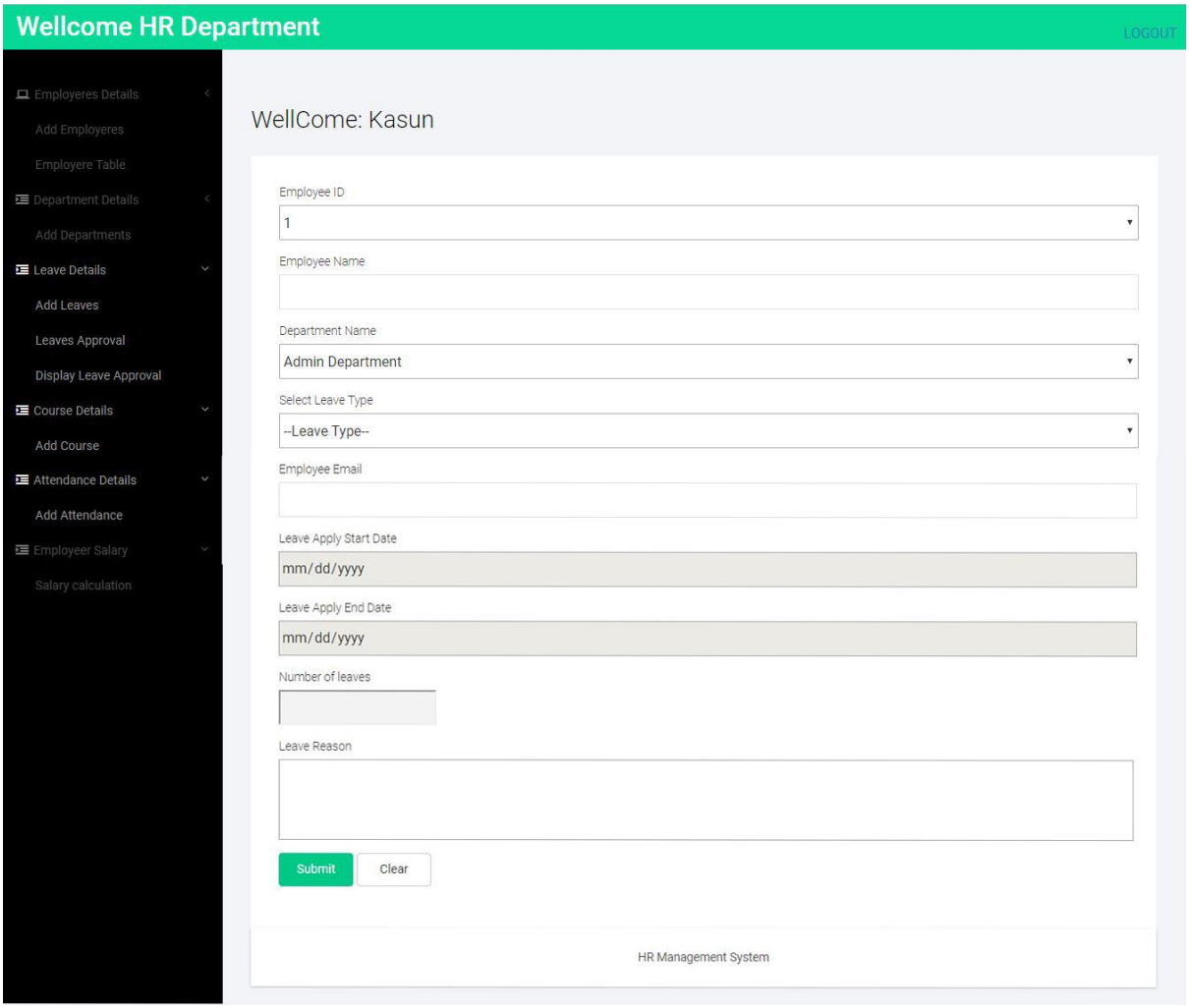
- **Admin**
- **Employee**
- **Customer**



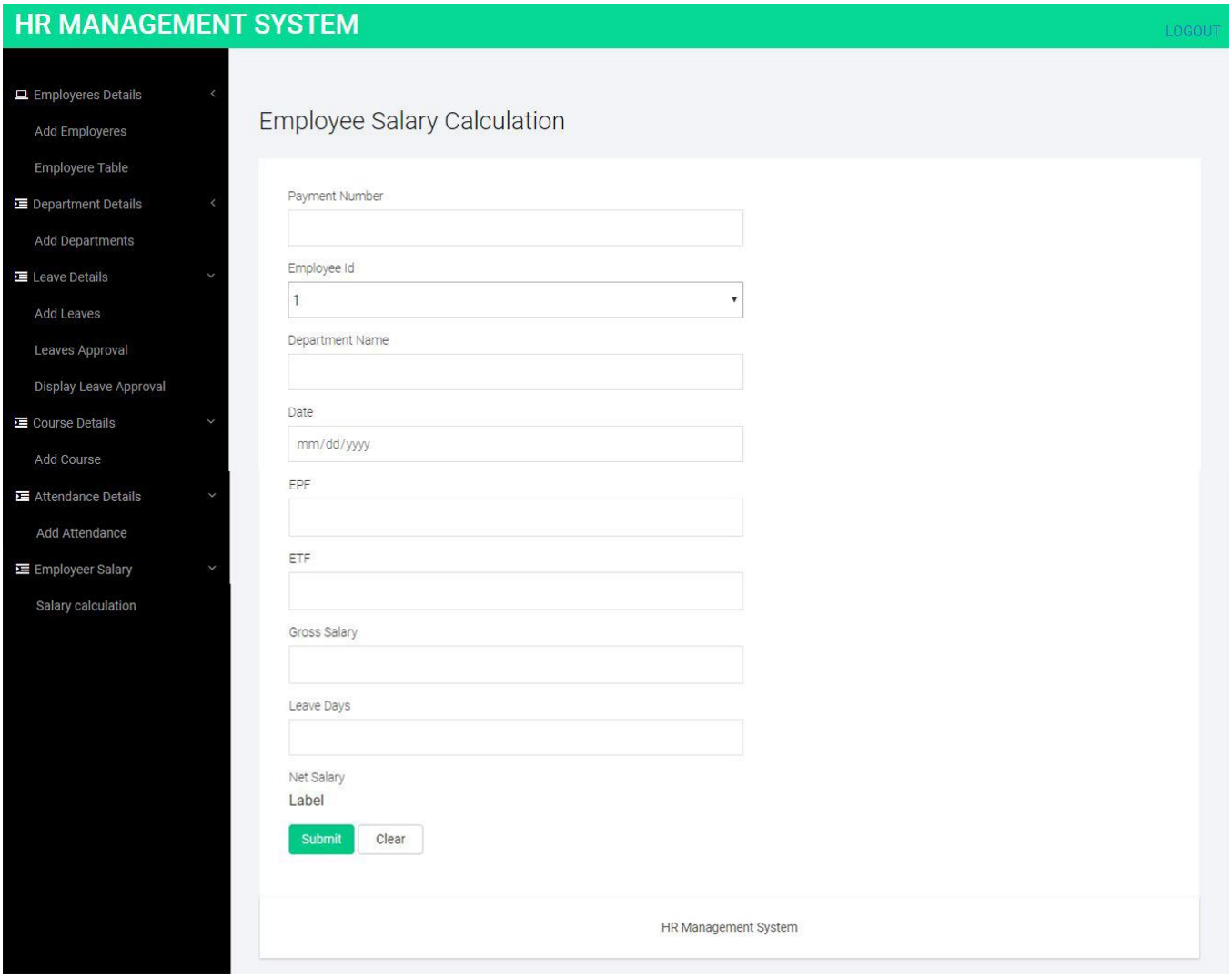
Employee Management



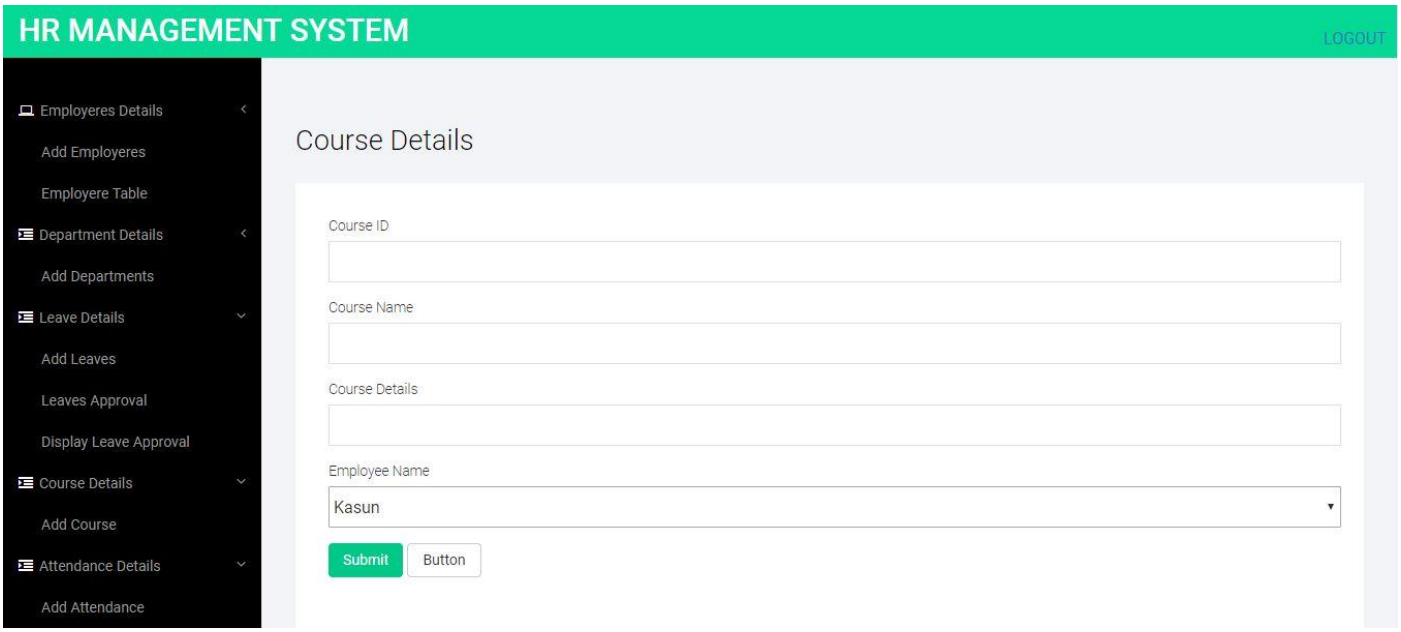
Leave Management



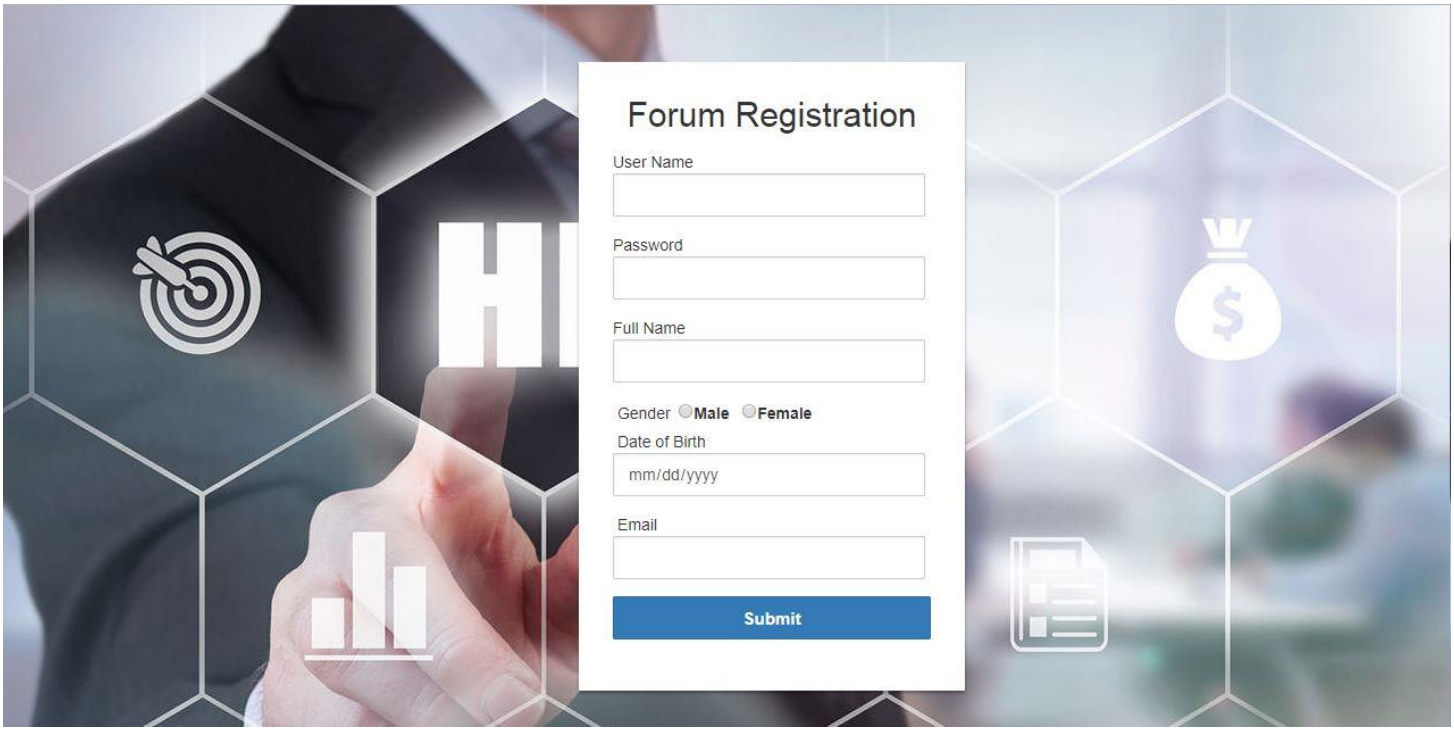
Payroll Management



Course Management



Course Forum



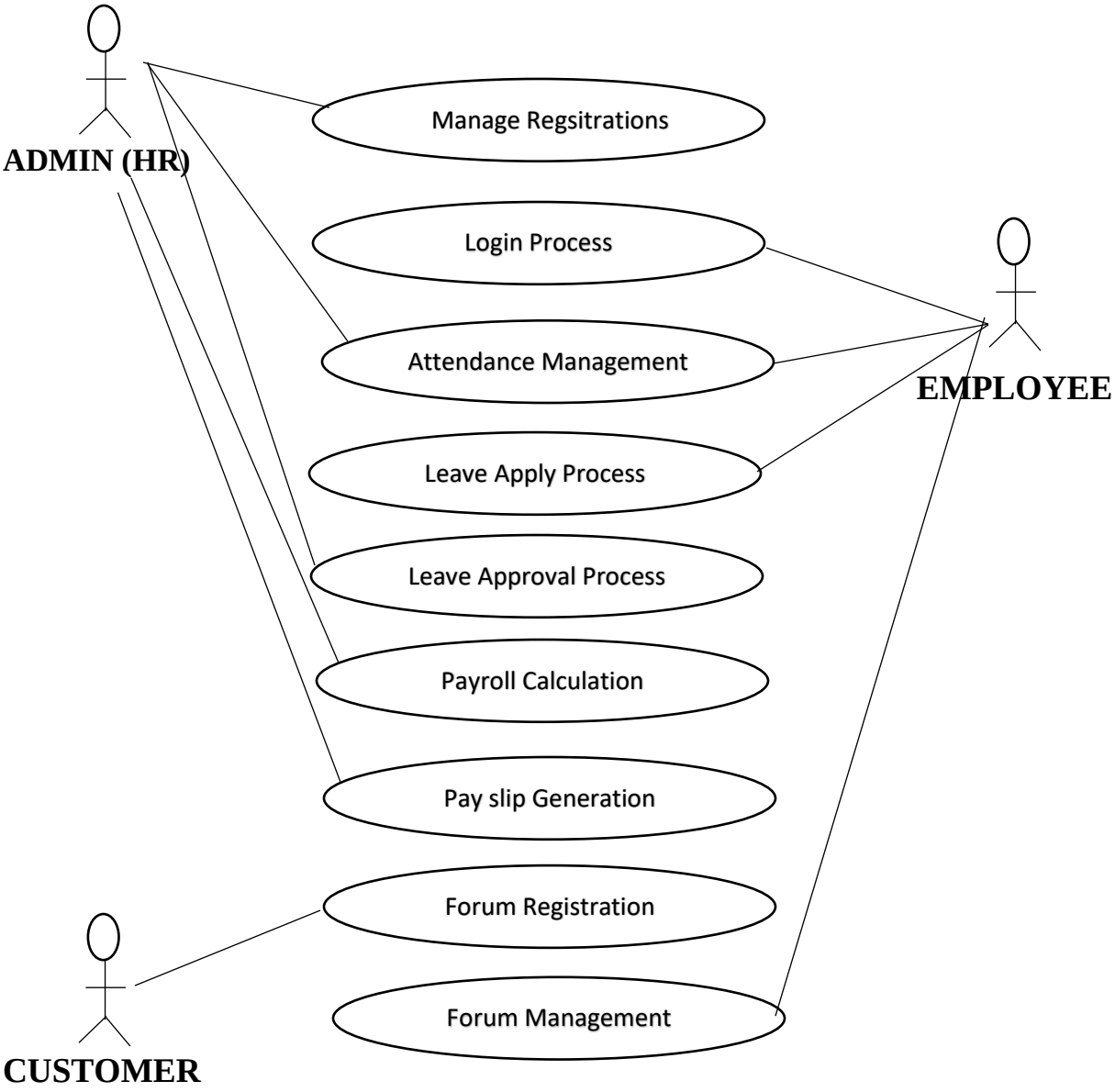
Description of Implemented functionalities

- **User Management** – The system allows users to first register to the system. There are different types of users:

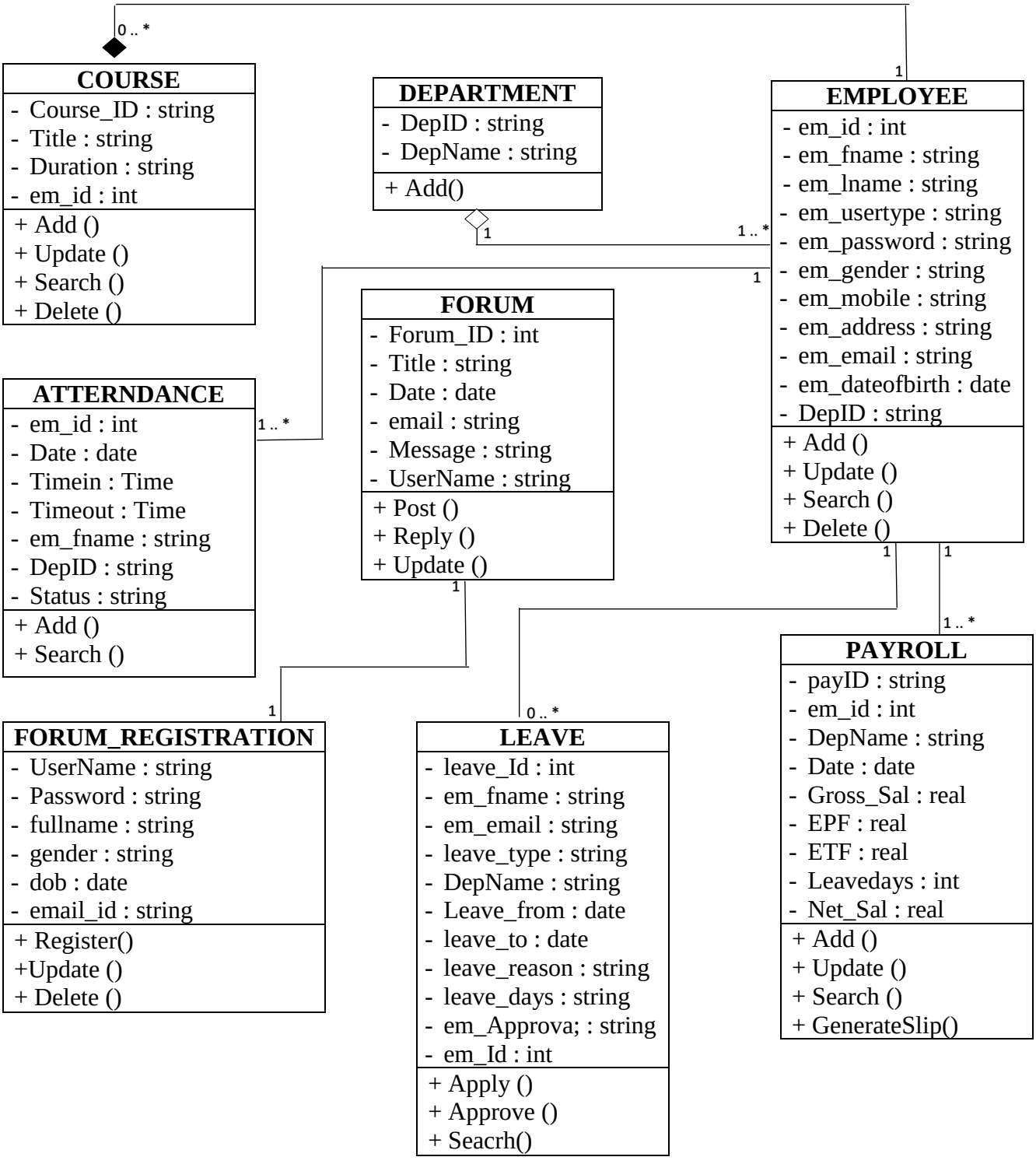
    - HR Admin - Who can do all operations on the system and with the data stored on the database.
    - Employee - Allowed only to apply leaves, record attendance time in and time out.
    - Customer – Allowed to view course details and register in to the Forum and Post an article.

- **Employee Management** – Allows to add an employee record, edit employee details, update details of employees, search employee details and delete an employee. The HR Admin is allowed to perform all the operations mentioned above.
- **Leave Management –** Allows an employee to apply a leave and the status will be sent to his/ her email.
- **Attendance Management –** All employees are allowed to record their time in and time out every day when they attend work.
- **Forum Management –** Allows a registered customer to post an article and the reply for these articles will be updated by an employee in the organization.
- **Course Management –** Allows to add a new course and its details, edit details, search details and delete a course.
- **Manage Departments -** Allows to add a department and delete.

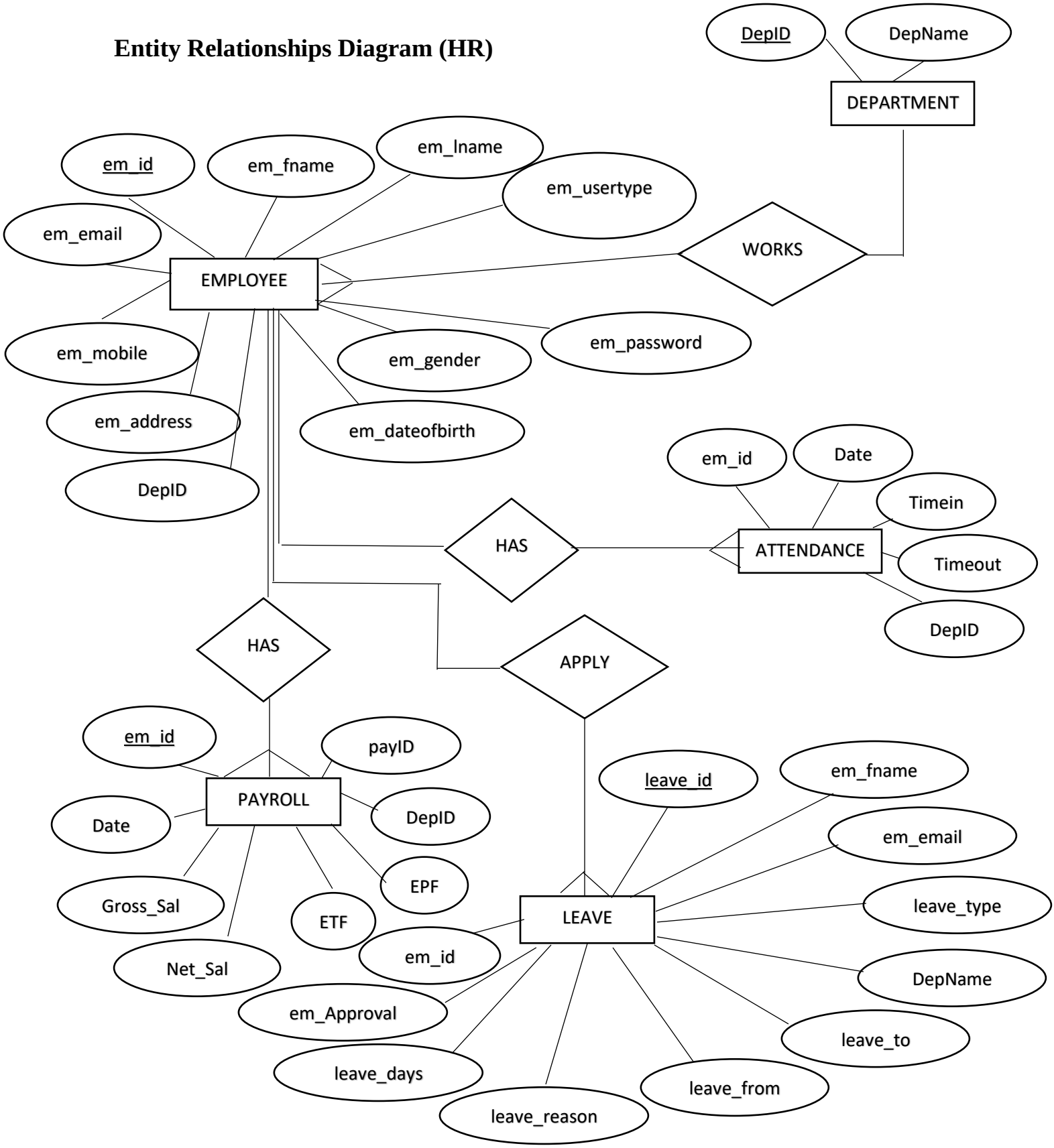
Use Case Diagram



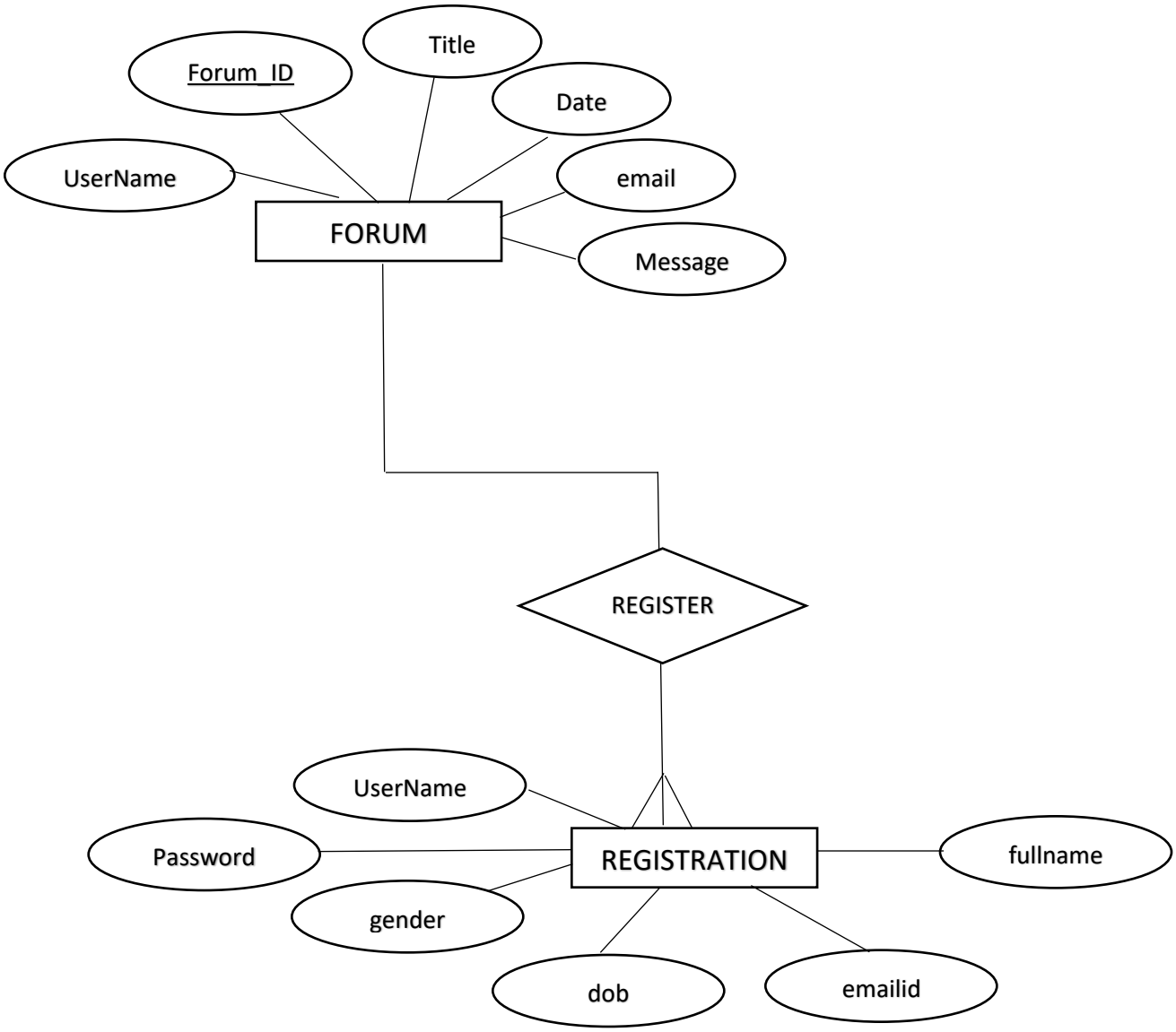
Class Diagram



Entity Relationships Diagram (HR)



Entity Relationships Diagram (Forum)



Algorithm

Calculate the Net Salary:

```
NetSal ← 0
EmpCount ← 1
WHILE EmpCount <= TotalEmplyees DO
     GET BasicSalary
     NoPay ← NoOfLeaveday * (BasicSalary / 30)
     EPF ← BasicSalary * 0.08
     ETF ← BasicSalary * 0.03
     NetSalary ← BasicSalary – NoPay – EPF – ETF
     OUTPUT NetSalary
END WHILE
```

Code Snippet:

```
            Ba = Convert.ToDouble(TextBox3.Text);
            Att = Convert.ToDouble(TextBox6.Text);
            EPF = Convert.ToDouble(TextBox4.Text);
            ETF = Convert.ToDouble(TextBox5.Text);
            EPF = Ba * 8 / 100;
            ETF = Ba * 3 / 100;

            total = Ba - EPF - ETF - ((Ba / 30) * Att);
```

Detail Description of the Classes and Some Methods Used:

User class - properties and methods description.

Properties	
Property	**Description**
`private string Username;`	Enter Employee Username
`private string Password;`	Enter Employee Password
Methods	
Method	**Description**
`public DataTable ValidateUser(User ob)`	Method for Select User information into the database.

Employee class - properties and methods description.

Properties	
Property	**Description**
`private int em_id;`	Unique ID allocated for the Employee
`private string em_fname;`	Fist Name of the Employee
`private string em_lname;`	Last Name of the Employee
`private string em_usertype;`	Type of of the Employee
`private string em_password;`	Password of the Employee
`private string em_gender;;`	Gender of the Employee
`private string em_mobile;`	Mobile of the Employee
`private string em_address;`	Address of the Employee
`private string em_email;`	Email of the Employee
`private string em_dateofbirth;`	Employee Date of Birth
`private string depID;`	Foreign ID allocated for the Department
Methods	
Method	**Description**
`public void Insert(Employer ob)`	Method for inserting Employee information into the database.
`public DataTable  getDeptDtails()` `public void deletestd(int em_Id)`	Get Department Id and Name. Delete Employee Details using ID

Department class - properties and methods description.

Properties	
Property	**Description**
`private string DepartmentId;`	Unique ID allocated for the Employee
`private string DepartmentName;`	Add Department Details.
Methods	
Method	**Description**
`public void Depadd(Department ob)`	Method for inserting Department information into the database.

Leave class - properties and methods description.

Properties	
Property	**Description**
`private int leave_Id;`	Unique leave ID allocated for the Employee.
`private string em_fname;`	Fist Name of the Employee
`private string em_email;`	Last Name of the Employee
`private string leave_type;`	Leave type of the Employee
`private string depID;`	Get employee Department Details.
`private string depName;`	Department of the Employee
`private string leave_to;`	Leave TO date of the Employee
`private string leave_from;`	Leave From date of the Employee
`private string leave_reason;`	Leave reason of the Employee
`private string leave_days;`	Employee leave days.
`private string em_Apporel;`	Employee Approval Details.
`private int em_id;`	Foreign ID allocated for the Employee Table.
Method	
Method	**Description**
`public void isertlevae(Leave ob)`	Method for inserting Employee Leave information into the database.
`public void updatelevae(Leave ob)`	Update Employee Details.

Employee Payment class - properties and methods description.

Properties	
Property	**Description**
`private string payID;`	Unique ID allocated for the Salary.
`private int em_Id;`	Get Employee Id from Leave table.
`private string DepName;`	Get employee Department from Leave table.
`private string Date;`	Select Date.
`private double Gross_Sal;`	Add Employee gross salary.
`private double EPF;`	Add Employee EPF Details.
`private double ETF;`	Add Employee ETF Details.
`private int leavesdays;`	Get Leave days from the leave table.
`private double Net_Sal;`	Auto calculates employee details.
Methods	
Method	**Description**
`public void isertsalary(Salary ob)`	Method for inserting Salary information into the database.

Employee Payment class - properties and methods description.

Properties	
Property	**Description**
`private string payID;`	Unique ID allocated for the Salary.
`private int em_Id;`	Get Employee Id from Leave table.
`private string DepName;`	Get employee Department from Leave table.
`private string Date;`	Select Date.
`private double Gross_Sal;`	Add Employee gross salary.
`private double EPF;`	Add Employee EPF Details.
`private double ETF;`	Add Employee ETF Details.
`private int leavesdays;`	Get Leave days from the leave table.
`private double Net_Sal;`	Auto calculates employee details.
Methods	
Method	**Description**
`public void isertsalary(Salary ob)`	Method for inserting Salary information into the database.

Course Details class - properties and methods description.

Properties	
Property	**Description**
`private int Cousrse_ID;`	Unique ID allocated for the Course.
`private string Title;`	Add Course table.
`private string Duratiom;`	Add Course Description.
`private int Em_Id;`	Get Employee Id from the Employee table.
Methods	
Method	**Description**
`public void Course(Courses ob)`	Method for inserting Course information into the database.

Employee

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Employer
    {
        private int em_id;
        private string em_fname;
        private string em_lname;
        private string em_usertype;
        private string em_password;
        private string em_gender;
        private string em_mobile;
        private string em_address;
        private string em_email;
        private string em_dateofbirth;
        private string depID;

        public int Em_id
        {
            get
            {
                return em_id;

            }
            set
            {
                em_id = value;
            }
        }
        public string Em_fname
        {
            get
            {
                return em_fname;

            }
            set
            {
                em_fname = value;
            }
        }
        public string Em_lname
        {
            get
            {
                return em_lname;


            }
            set
            {
                em_lname = value;
            }
```

```
}
public string Em_usertype
{
    get
    {
        return em_usertype;


    }
    set
    {
        em_usertype = value;
    }
}
 public string Em_password
{
    get
    {
        return em_password;


    }
    set
    {
        em_password = value;
    }}
 public string Em_gender
{
    get
    {
        return em_gender;


    }
    set
    {
        em_gender = value;
    }
       }
public string Em_mobile
{
    get
    {
        return em_mobile;


    }
    set
    {
        em_mobile = value;
    }
}
public string Em_address
{
    get
    {
        return em_address;


    }
    set
    {
        em_address = value;
```

```
            }
        }
        public string Em_email
        {
            get
            {
                return em_email;


            }
            set
            {
                em_email = value;
            }
        }

        public string Em_dateofbirth
        {
            get
            {
                return em_dateofbirth;


            }
            set
            {
                em_dateofbirth = value;
            }
        }
        public string DepID
        {
            get
            {
                return depID;


            }
            set
            {
                depID = value;
            }
        }
        public Employer(string em_fnames, string em_lnames, string em_usertypes, string
em_passwords, string em_genders, string em_mobiles, string em_addresss, string em_emails,
string em_dateofbirths, string DId)
        {
            this.em_fname = em_fnames;
            this.em_lname = em_lnames;
      this.em_usertype = em_usertypes;
    this.em_password = em_passwords;
    this.em_gender = em_genders;
            this.em_mobile = em_mobiles;
            this.em_address = em_addresss;
            this.em_email = em_emails;
          this.em_dateofbirth = em_dateofbirths;
          this.depID = DId;

        }


        public Employer(int em_ids, string em_fnames, string em_lnames, string
em_usertypes, string em_passwords, string em_genders, string em_mobiles, string
em_addresss, string em_emails, string em_dateofbirths)
```

```
        {
            this.em_id = em_ids;
            this.em_fname = em_fnames;
            this.em_lname = em_lnames;
            this.em_usertype = em_usertypes;
            this.em_password = em_passwords;
            this.em_gender = em_genders;
            this.em_mobile = em_mobiles;
            this.em_address = em_addresss;
            this.em_email = em_emails;
            this.em_dateofbirth = em_dateofbirths;
        }
        public Employer(int em_ids)
        {
            this.em_id = em_ids;

        }
    }
                                            }
```

Login

```
       public string em_fname { get; set; }
        public string em_password { get; set; }
        public string UserName { get; set; }
        public string Password { get; set; }
```

Salary

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Salary
    {
        private string payID;
        private int em_Id;
        private string DepName;
        private string Date;
        private double Gross_Sal;
        private double EPF;
        private double ETF;
        private int leavesdays;
        private double Net_Sal;

        public string PayID
        {
            get
            {
                return payID;


            }
            set
            {
                payID = value;
            }
        }
```

```
public int Em_Id
{
    get
    {
        return em_Id;


    }
    set
    {
        em_Id = value;
    }
}
public string depName
{
    get
    {
        return DepName;


    }
    set
    {
        DepName = value;
    }
}
public string date
{
    get
    {
        return Date;


    }
    set
    {
        Date = value;
    }
}
public double gross_Sal
{
    get
    {
        return Gross_Sal;


    }
    set
    {
        Gross_Sal = value;
    }
}
public double epf
{
    get
    {
        return EPF;


    }
    set
    {
        EPF = value;
```

```
            }
        }
        public double etf
        {
            get
            {
                return ETF;


            }
            set
            {
                ETF = value;
            }
        }
        public int Leavesdays
        {
            get
            {
                return leavesdays;


            }
            set
            {
                leavesdays = value;
            }
        }
        public double net_Sal
        {
            get
            {
                return Net_Sal;


            }
            set
            {
                Net_Sal = value;
            }
        }
        public Salary(string PayIDs, int Em_Ids, string DepNames, string dates, double
gross_Sals, double epfs, double etfs, int Leavesdays, double net_Sals)
        {

            this.PayID = PayIDs;
            this.Em_Id = Em_Ids;
            this.DepName = DepNames;
            this.date = dates;
            this.gross_Sal = gross_Sals;
            this.epf = epfs;
            this.etf = etfs;
            this.Leavesdays = Leavesdays;
            this.net_Sal = net_Sals;
        }
    }
                                            }
```

Department

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Department
    {
        private string DepartmentId;
        private string DepartmentName;



        public string departmentId
        {
            get
            {
                return DepartmentId;


            }
            set
            {
                DepartmentId = value;
            }
        }
        public string departmentName
        {
            get
            {
                return DepartmentName;


            }
            set
            {
                DepartmentName = value;
            }
        }
        public Department(string DepartmentId, string DepartmentName)
        {

            this.DepartmentId = DepartmentId;
            this.DepartmentName = DepartmentName;

        }
    }
                                            }
```

Salary Calculation Part

```
protected void Button1_Click(object sender, EventArgs e)
        {
             try
            {
            double Ba, EPF, ETF, Att, total;
            Ba = Convert.ToDouble(TextBox3.Text);
            Att = Convert.ToDouble(TextBox6.Text);
            EPF = Convert.ToDouble(TextBox4.Text);
            ETF = Convert.ToDouble(TextBox5.Text);
            EPF = Ba * 8 / 100;
            ETF = Ba * 3 / 100;

            total = Ba - EPF - ETF - ((Ba / 30) * Att);
            Label1.Text = total.ToString("");



            Salary sob = new Salary(TextBox7.Text,
Convert.ToInt32(DropDownList1.SelectedValue.ToString()), TextBox1.Text, TextBox2.Text,
Convert.ToDouble(TextBox3.Text), Convert.ToDouble(TextBox4.Text),
Convert.ToDouble(TextBox5.Text), Convert.ToInt32(TextBox6.Text),
Convert.ToDouble(Label1.Text));
            Class1 dbcon = new Class1();
            dbcon.isertsalary(sob);

            Response.Write("<script LANGUAGE='JavaScript' >alert('Save
Successful')</script>");
            TextBox2.Text = string.Empty;
            TextBox3.Text = string.Empty;
            }
             catch (Exception ex)
             {

             }
             finally
             {
```

Login (HR System)

```
        //Create user login
        public DataTable ValidateUser(User ob)
        {

            try
            {
                mcon.Open();
                DataTable dt = new DataTable();
                SqlCommand cmd = new SqlCommand("select * from Employee where em_fname='"
+ ob.em_fname + "'and em_password='" + ob.em_password + "'", mcon);

                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                sda.Fill(dt);
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.Read())
                {
                    publicVariables.Username = dr[1].ToString();
                    publicVariables.EmployeeID = Convert.ToInt32(dr[0]);
                }
                return dt;
            }

            finally
            {
                mcon.Close();
            }
```

Login (Forum)

```
       //Create forum login
        public DataTable fouremUser(User ob)
        {

            try
            {
                mcon.Open();
                DataTable dt = new DataTable();
                SqlCommand cmd = new SqlCommand("select * from Forum_Registration where
UserName='" + ob.UserName + "'and Password='" + ob.Password + "'", mcon);
                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                sda.Fill(dt);
                return dt;
            }

            finally
            {
                mcon.Close();
            }
```

Employee Handling

```
//Create insert emplyeer
        public void Insert(Employer ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into
Employee(em_fname,em_lname,em_usertype,em_password,em_gender,em_mobile,em_address,em_emai
l,em_dateofbirth,DepID) values ('" + ob.Em_fname + "','" + ob.Em_lname + "','" +
ob.Em_usertype + "','" + ob.Em_password + "','" + ob.Em_gender + "','" + ob.Em_mobile +
"','" + ob.Em_address + "','" + ob.Em_email + "','" + ob.Em_dateofbirth + "','" +
ob.DepID + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
```

Department Handling

```
 public void Depadd(Department ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into Department(DepID,DepName) values ('" +
ob.departmentId + "','" + ob.departmentName + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
```

Attendance Handling

```
        public void AddAtteand(EmAttendance ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into
Attendance(em_Id,Date,Timein,Timeout,em_fname,DepID,Status) values ('" + ob.Em_id + "','"
+ ob.date + "','" + ob.timein + "','" + ob.timeout + "','" + ob.em_fname + "','" +
ob.DepID + "','" + ob.Status + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
```

Payroll

```
        public void isertsalary(Salary ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into
Payroll(payID,em_Id,DepName,Date,Gross_Sal,EPF,ETF,Leavedays,Net_Sal) values('" +
ob.PayID + "','" + ob.Em_Id + "','" + ob.depName + "','" + ob.date + "','" + ob.gross_Sal
+ "','" + ob.epf + "','" + ob.etf + "','" + ob.Leavesdays + "','" + ob.net_Sal + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
```

Searching

```
   public void findmemberByIds(int Id)
        {

            DataTable dt = new DataTable();
            SqlCommand cmd = new SqlCommand("select em_fname from Employee where  em_Id
='" + Id + "'", mcon);
            mcon.Open();
            //sqldat = new SqlCommand(sqlQ, mcon);
            //SqlDataAdapter sda = new SqlDataAdapter(cmd);
            SqlDataReader dr = cmd.ExecuteReader();

            if (dr.Read())
            {
                emfname = dr[0].ToString();

            }
```

Leave Process

```
        public void isertlevae(Leave ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into
Leave(em_fname,em_email,leave_type,DepName,leave_to,leave_from,leave_reason,leave_days,em
_Id) values('" + ob.Em_fname + "','" + ob.Em_email + "','" + ob.Leave_type + "','" +
ob.DepName + "','" + ob.Leave_to + "','" + ob.Leave_from + "','" + ob.Leave_reason +
"','" + ob.Leave_days + "','" + ob.Em_id + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
        public void updatelevae(Leave ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "update Leave set em_Apporel='" + ob.Em_Apporel + "' where
leave_Id ='" + ob.Leave_Id + "'";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
```

```
  }
            finally
            {
                mcon.Close();
            }
        }
```

Forum

```
  public void Foureminsert(Forumregister ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into
Forum_Registration(UserName,Password,fullname,gender,dob,emailid) values ('" +
ob.userName + "','" + ob.password + "','" + ob.fullname + "','" + ob.gender + "','" +
ob.dob + "','" + ob.emailid + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }

        public void Fouremmessage(Forumfeedback ob)
        {
            try
            {
                mcon.Open();
                string sqlQ = "insert into Forum(Title,Date,email,Message,UserName)
values ('" + ob.title + "','" + ob.date + "','" + ob.email + "','" + ob.message + "','" +
ob.userName + "')";
                SqlCommand cmd = new SqlCommand(sqlQ, mcon);
                cmd.ExecuteNonQuery();

            }
            catch (Exception ex)
            {

            }
            finally
            {
                mcon.Close();
            }
        }
```

User Manual

1. User Registration:

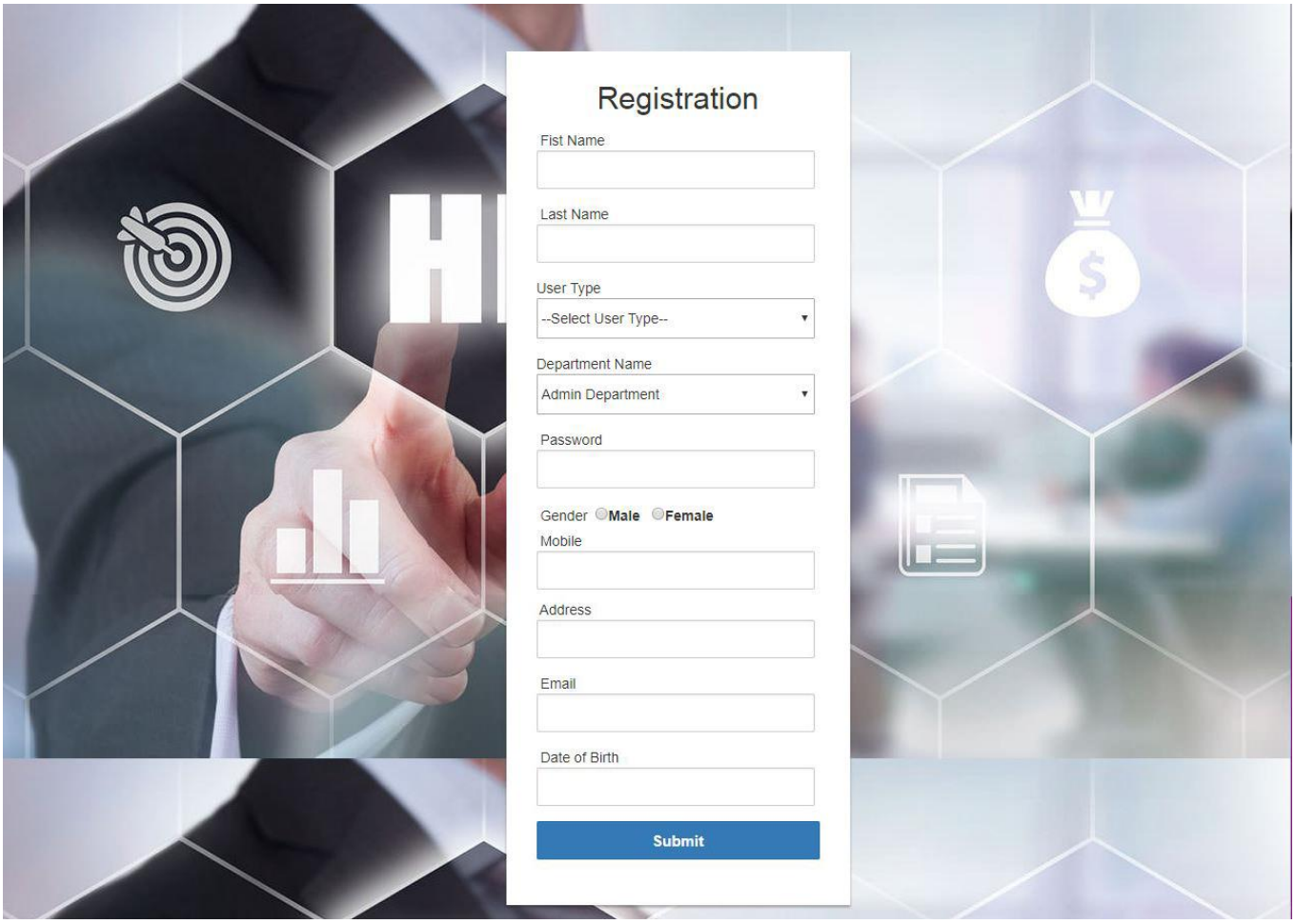


Figure 1.1

The beginning point of the system, where the user registrations take place in order to use the system. There are 3 types of users:

1. The Administrator
2. The Employee
3. The Customer

Login:

Figure 12

The registered user enters the username and the password to be validated to start using the functions of the system.

Reject Invalid Username and Password

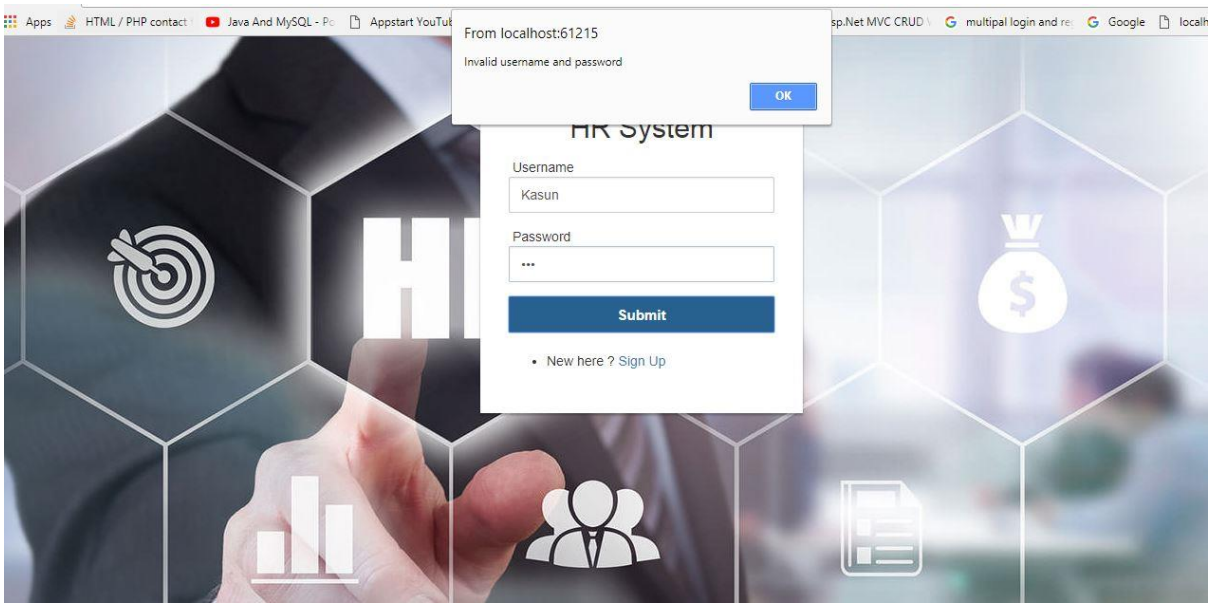


Figure 1.3

2. The HR User Profile for the HR Department:

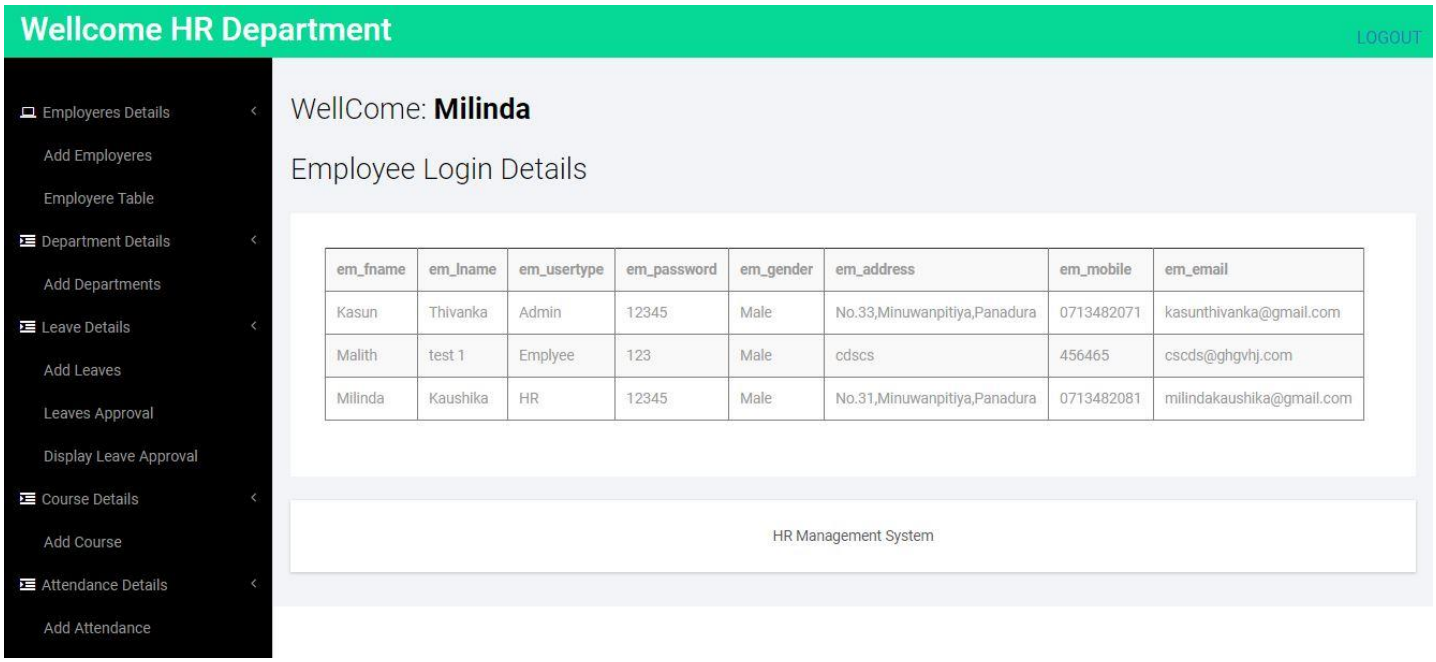


Figure 2.1

Search a User Profile

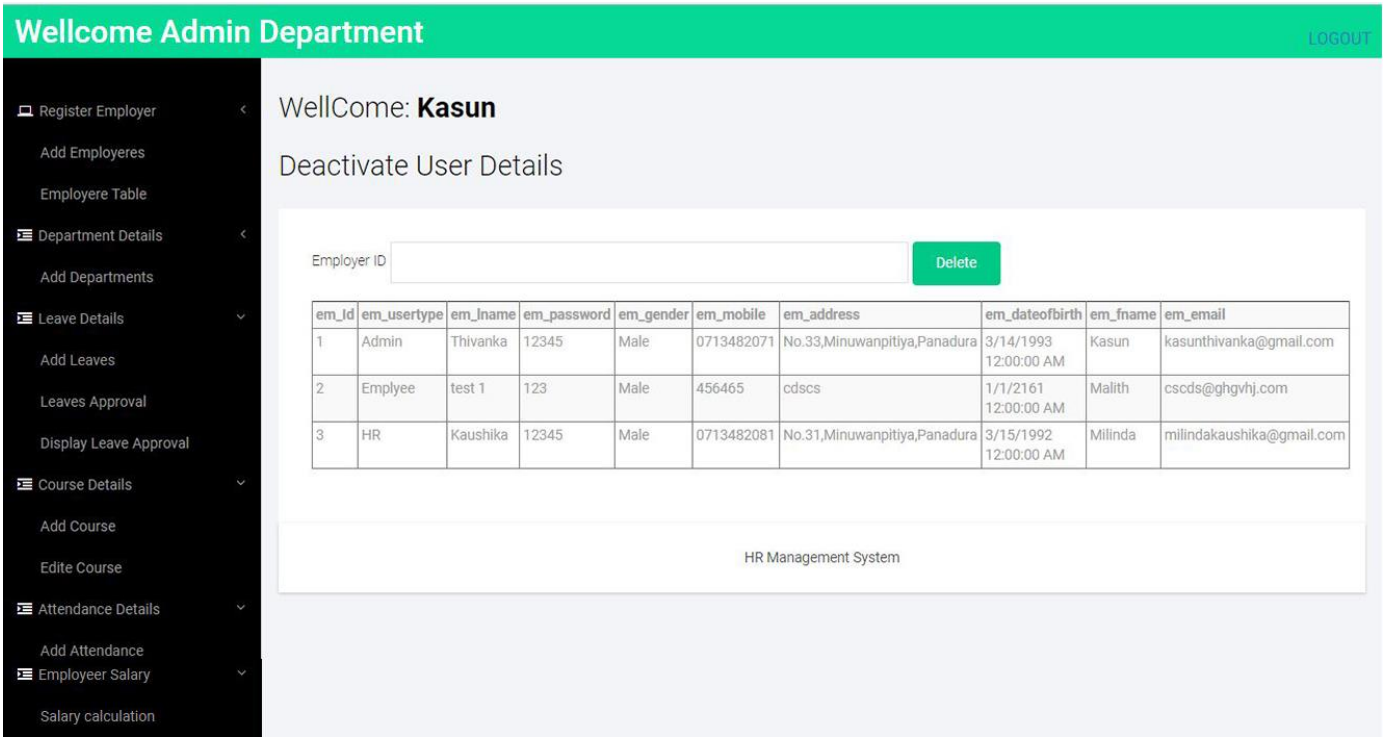


Figure 2.2

3. Add an Employee Record to the system database:

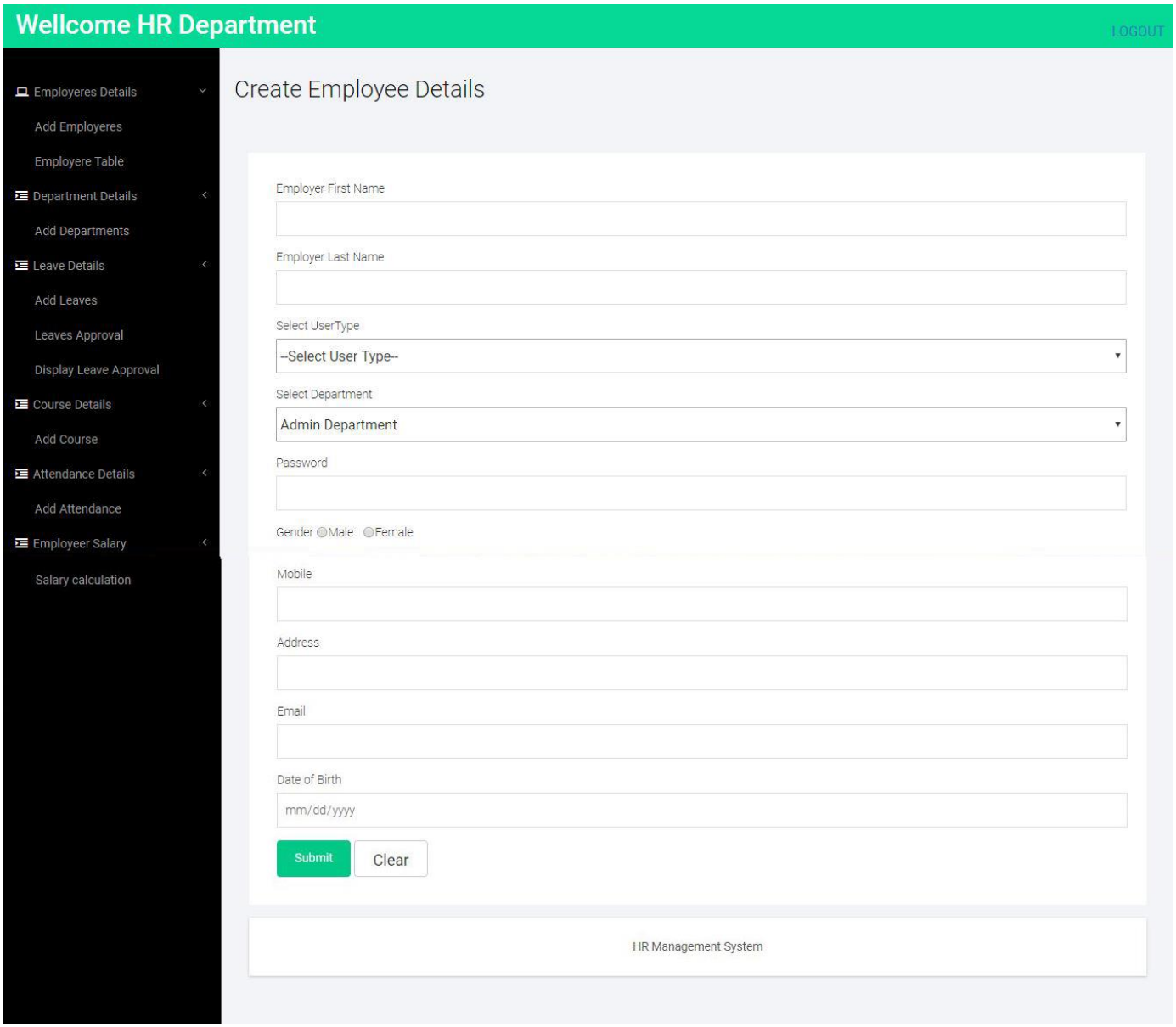


Figure 3.1

Edit an Employee Detail from the system database

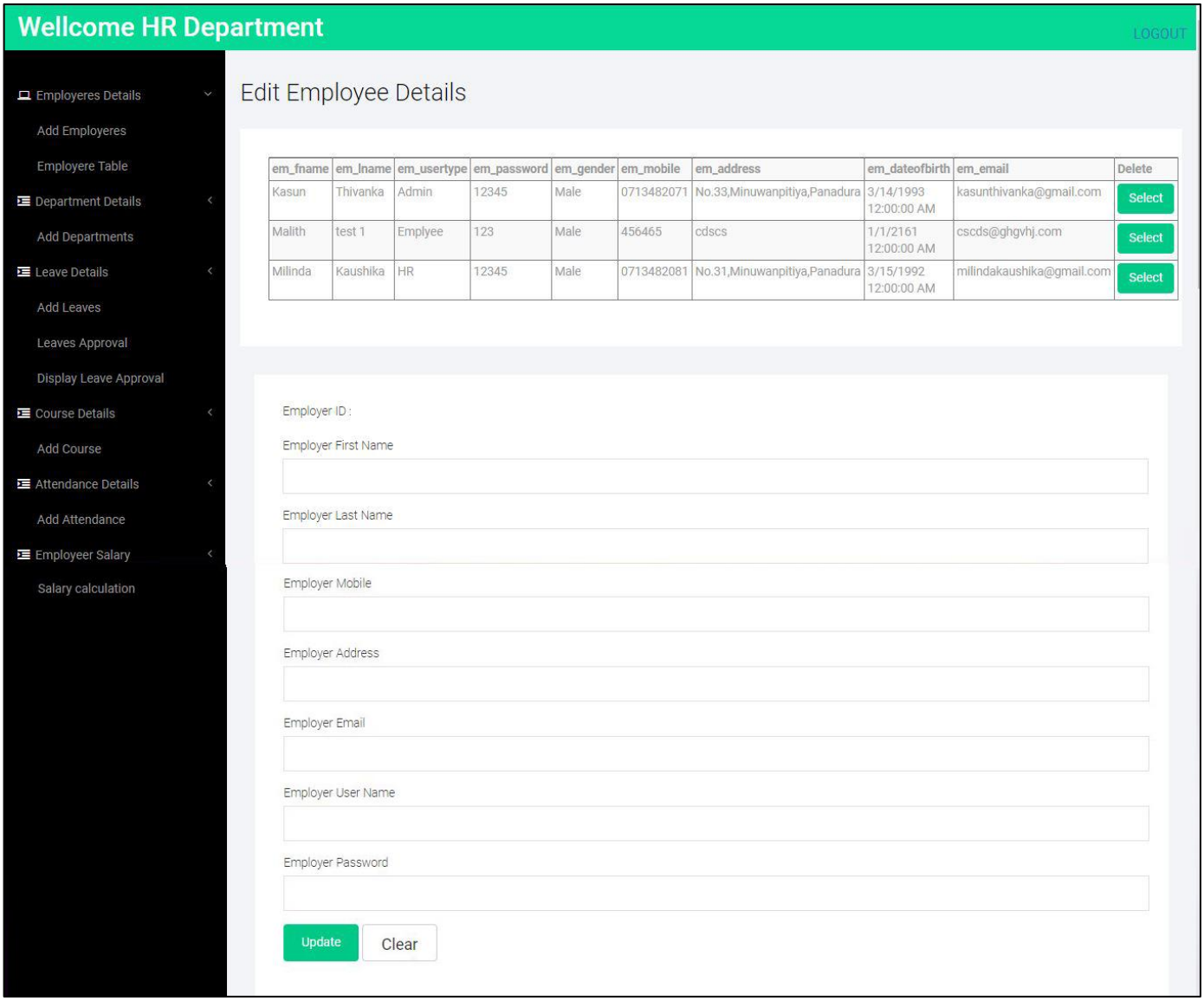


Figure 3.2

4.Add a New Department to the system database

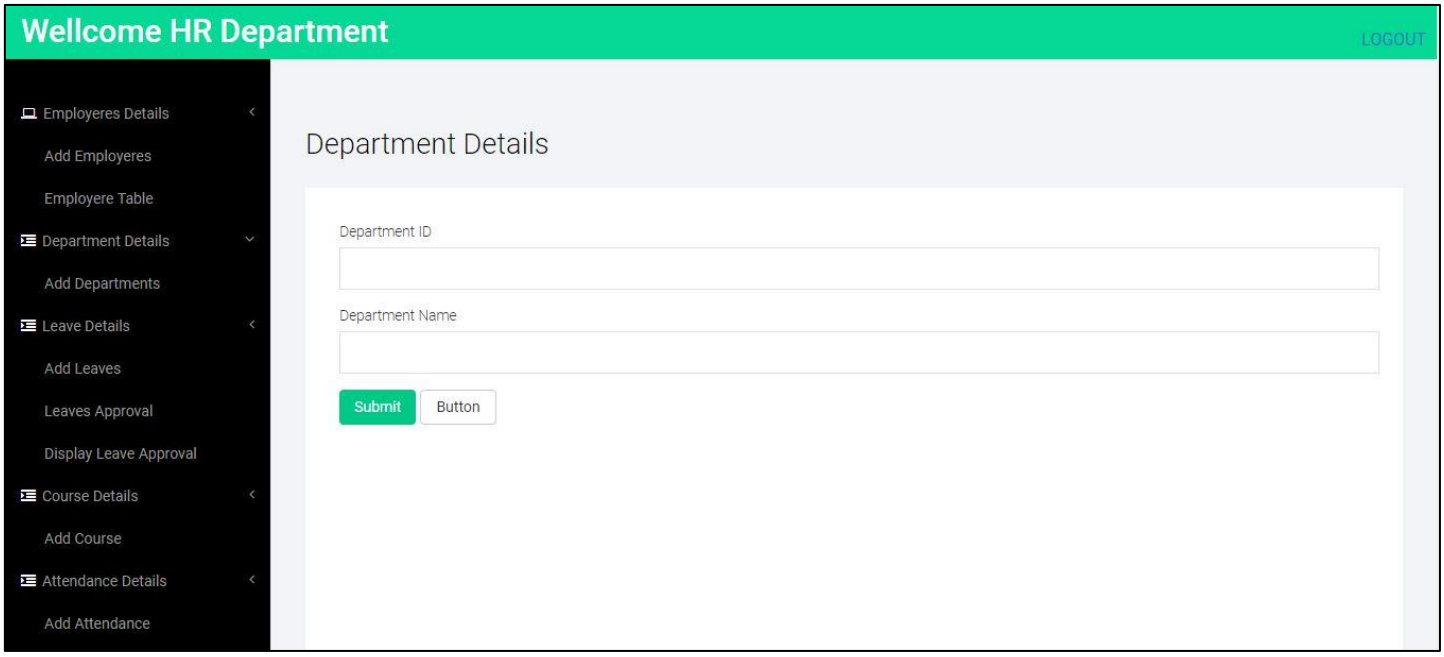


Figure4.1

5. Apply a Leave

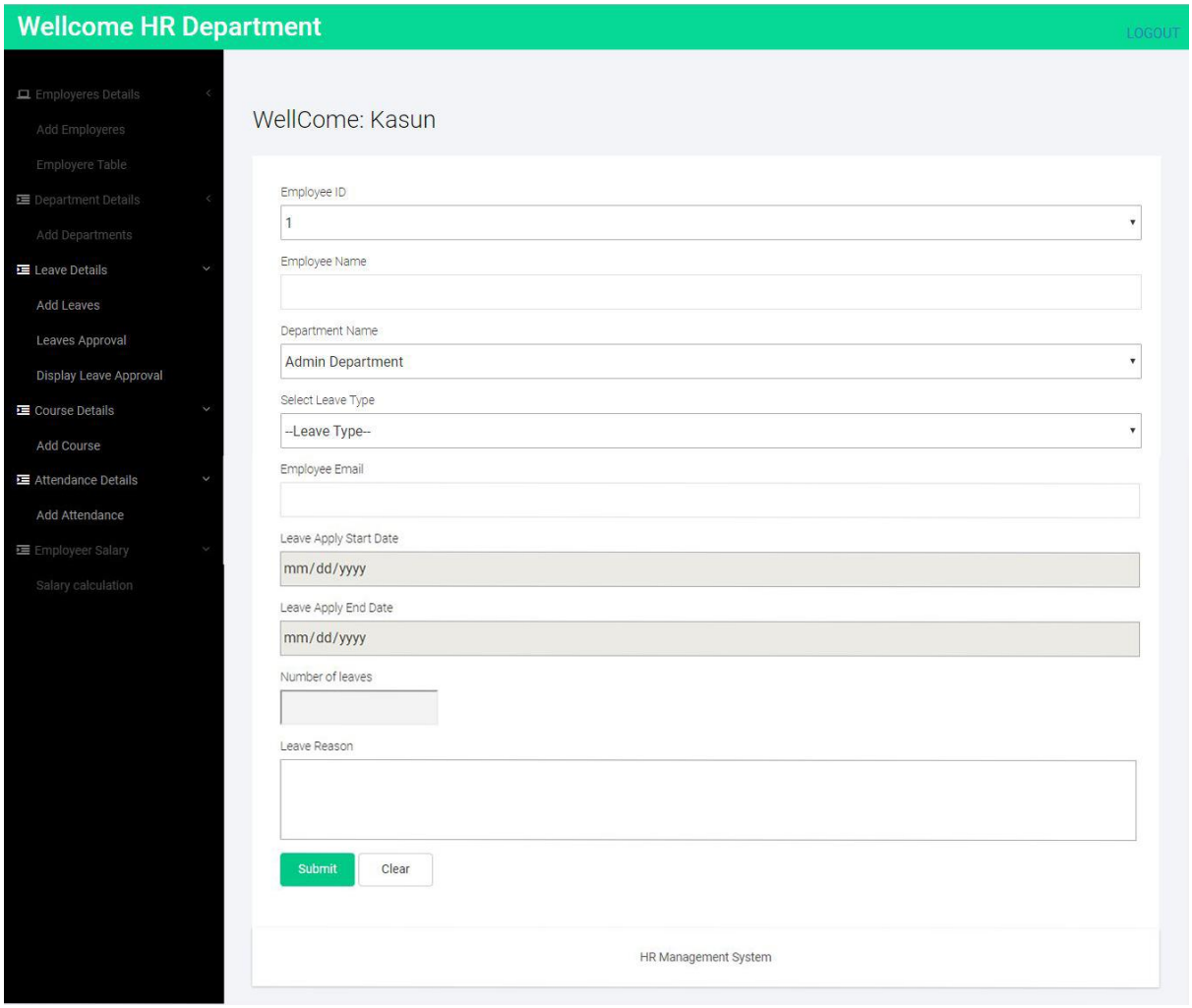
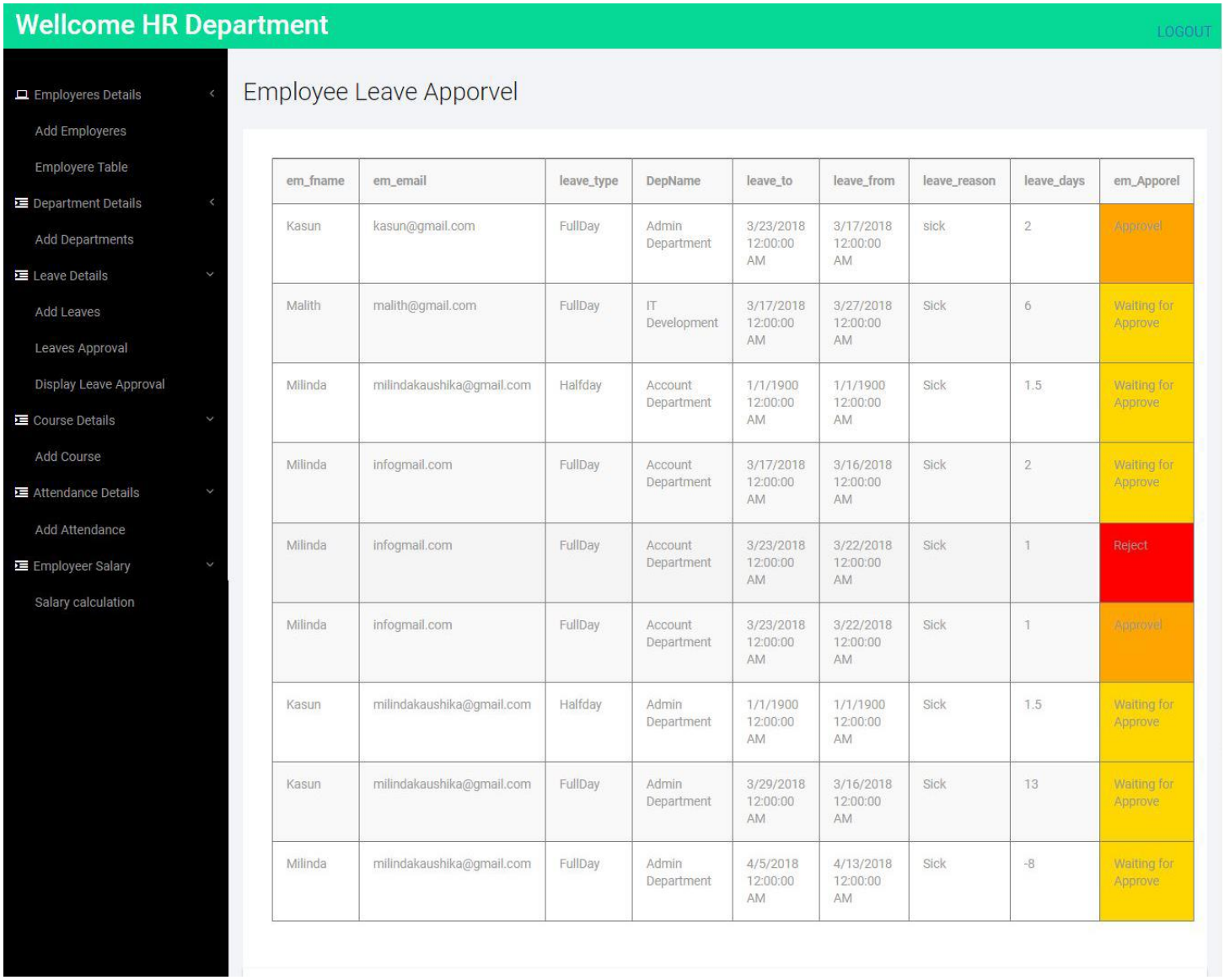


Figure5.1

View Applied Leaves Detail



em_fname	em_email	leave_type	DepName	leave_to	leave_from	leave_reason	leave_days	em_Apporel
Kasun	kasun@gmail.com	FullDay	Admin Department	3/23/2018 12:00:00 AM	3/17/2018 12:00:00 AM	sick	2	Approvel
Malith	malith@gmail.com	FullDay	IT Development	3/17/2018 12:00:00 AM	3/27/2018 12:00:00 AM	Sick	6	Waiting for Approve
Milinda	milindakaushika@gmail.com	Halfday	Account Department	1/1/1900 12:00:00 AM	1/1/1900 12:00:00 AM	Sick	1.5	Waiting for Approve
Milinda	infogmail.com	FullDay	Account Department	3/17/2018 12:00:00 AM	3/16/2018 12:00:00 AM	Sick	2	Waiting for Approve
Milinda	infogmail.com	FullDay	Account Department	3/23/2018 12:00:00 AM	3/22/2018 12:00:00 AM	Sick	1	Reject
Milinda	infogmail.com	FullDay	Account Department	3/23/2018 12:00:00 AM	3/22/2018 12:00:00 AM	Sick	1	Approvel
Kasun	milindakaushika@gmail.com	Halfday	Admin Department	1/1/1900 12:00:00 AM	1/1/1900 12:00:00 AM	Sick	1.5	Waiting for Approve
Kasun	milindakaushika@gmail.com	FullDay	Admin Department	3/29/2018 12:00:00 AM	3/16/2018 12:00:00 AM	Sick	13	Waiting for Approve
Milinda	milindakaushika@gmail.com	FullDay	Admin Department	4/5/2018 12:00:00 AM	4/13/2018 12:00:00 AM	Sick	-8	Waiting for Approve

Figure 5.2

8. Approval of a Leave

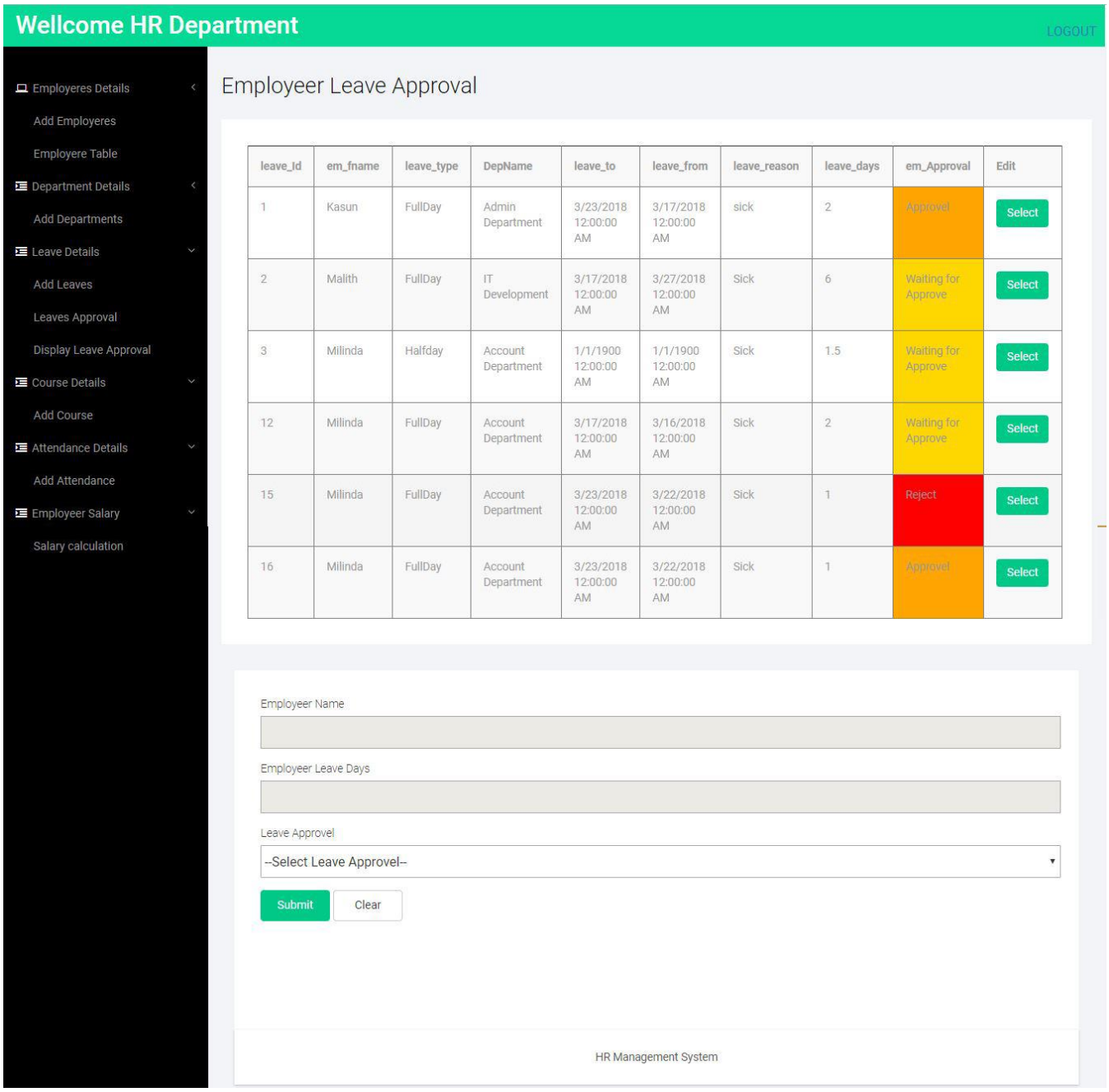


Figure 5.3

6. Salary Calculation

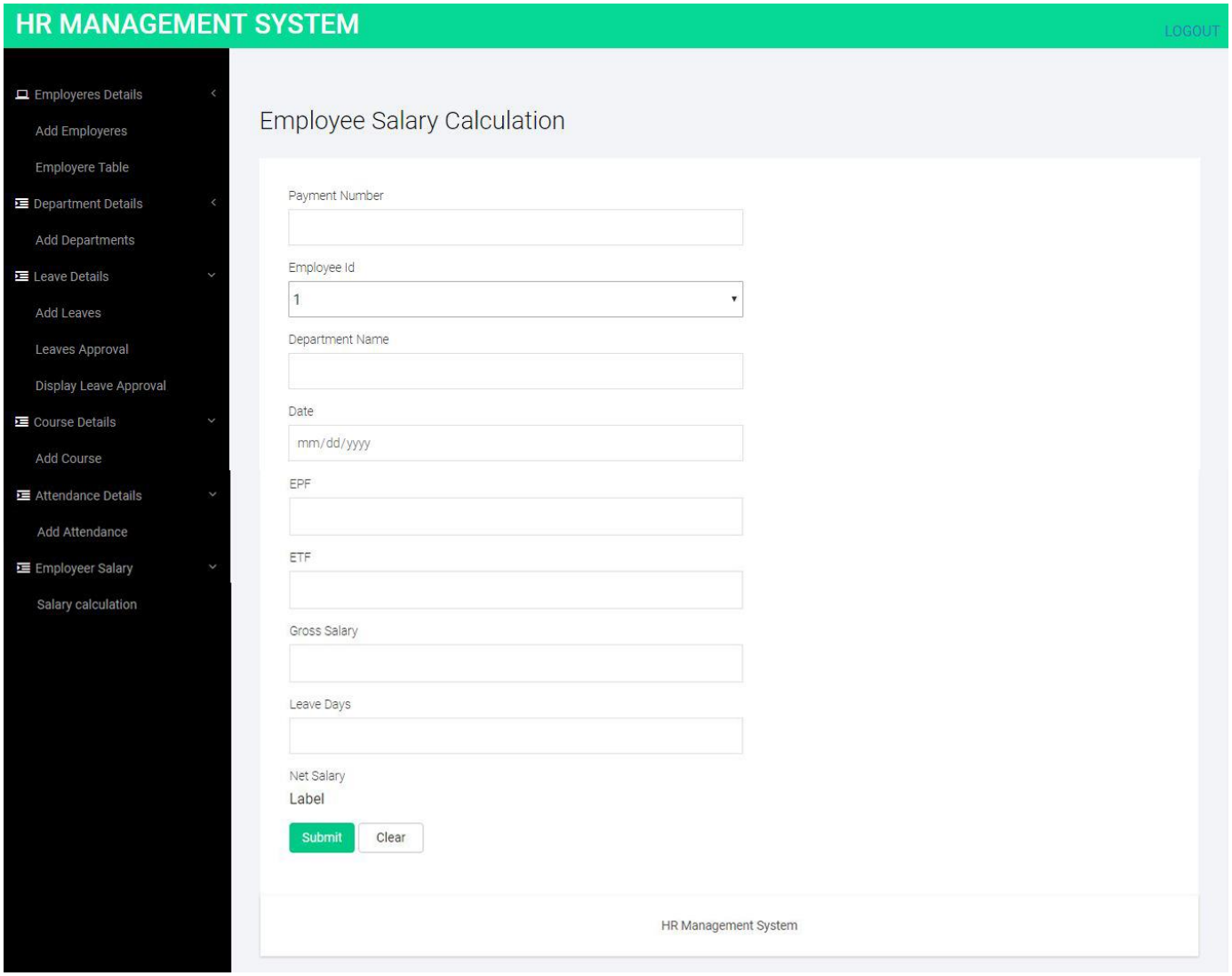


Figure 6.1

7. Forum Registration

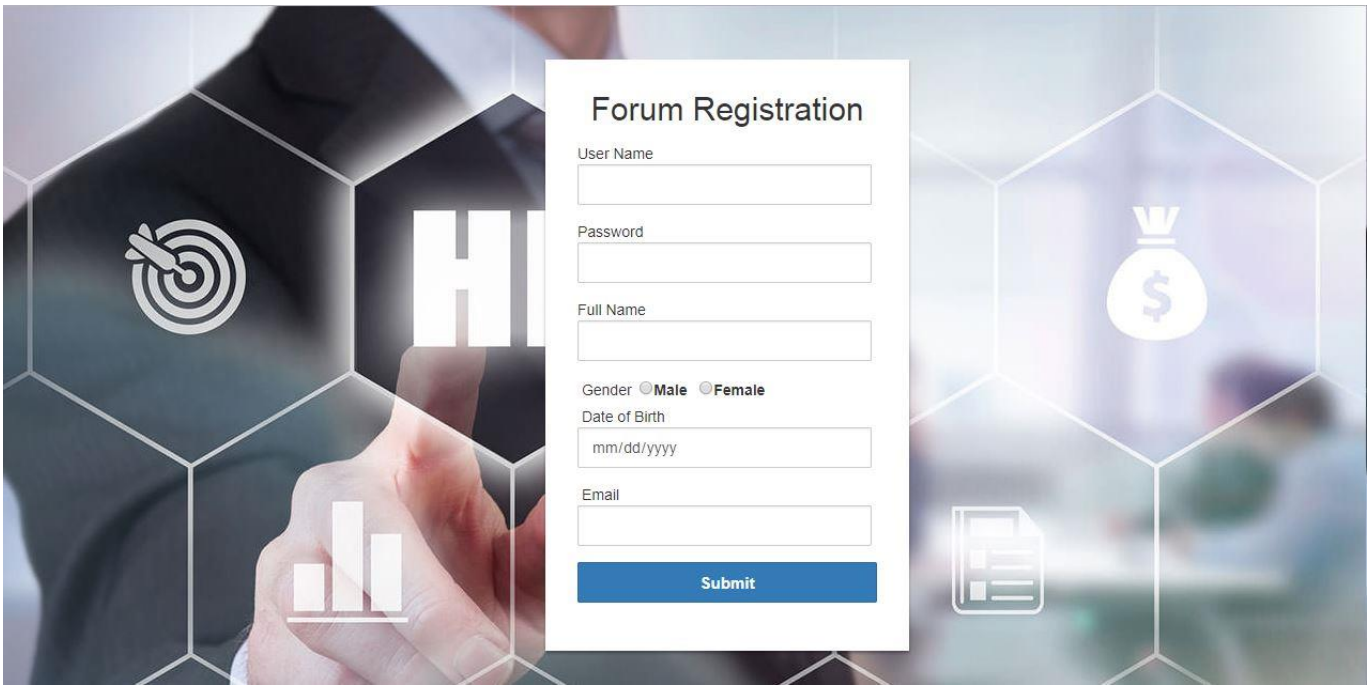


Figure 7.1

Login

Figure 7.2

Add a Post

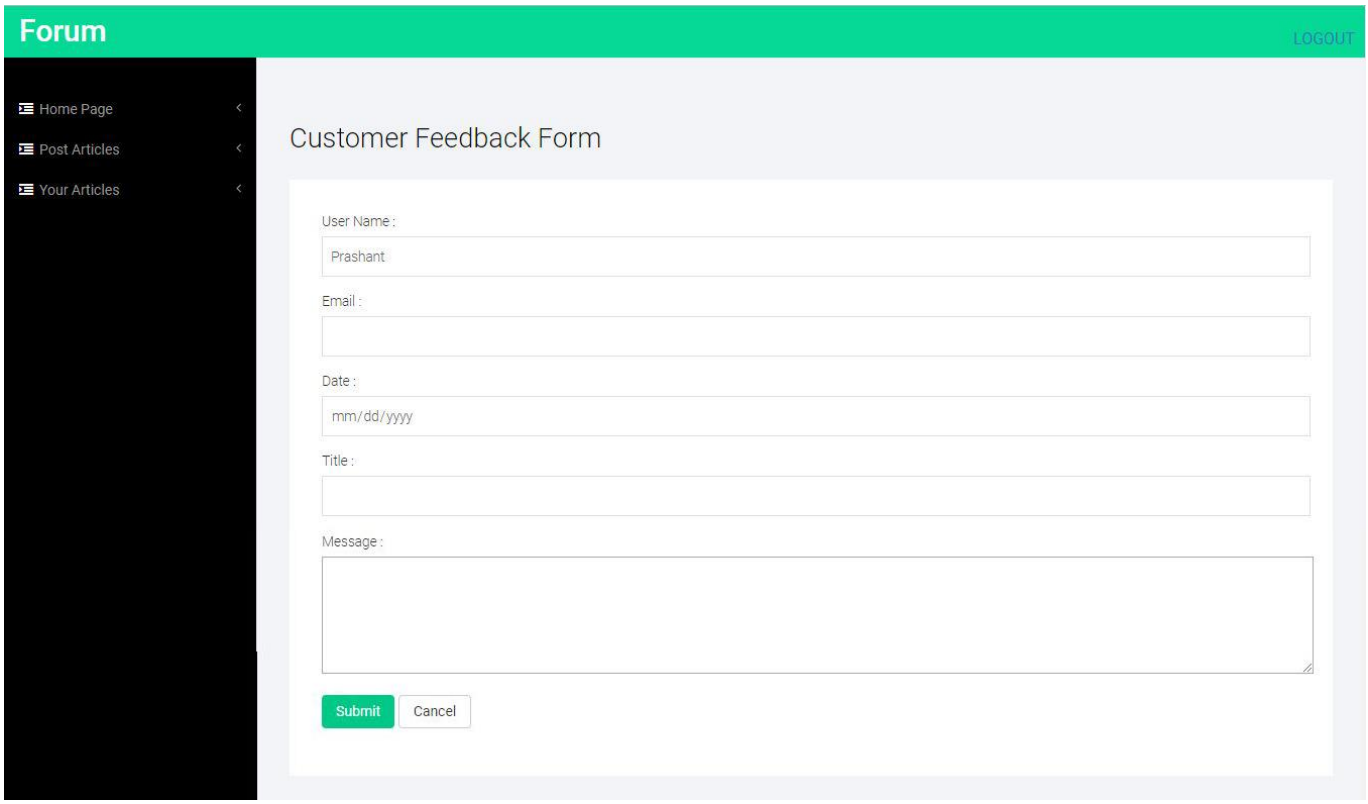


Figure 7.3

Posted Articles

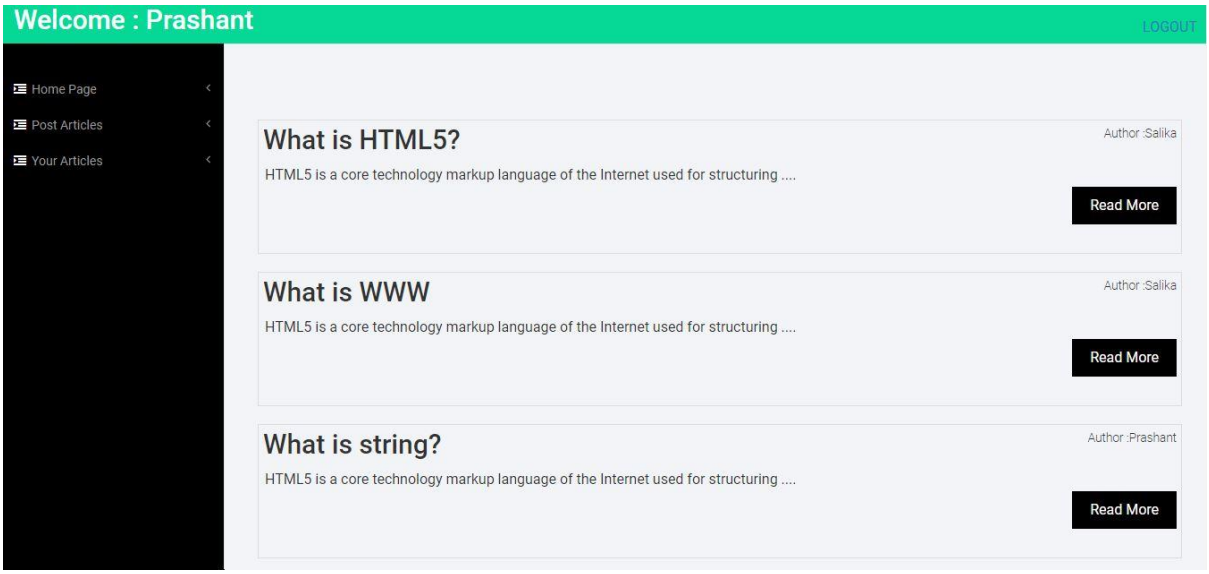


Figure 7.4

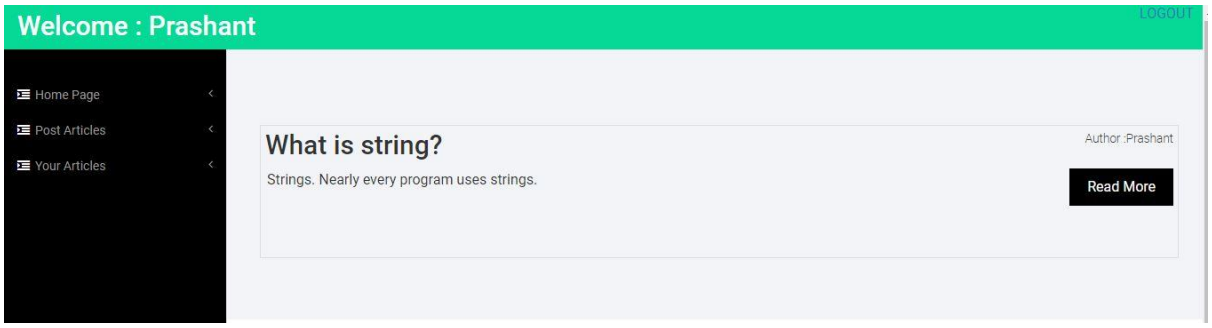


Figure7.5

Database

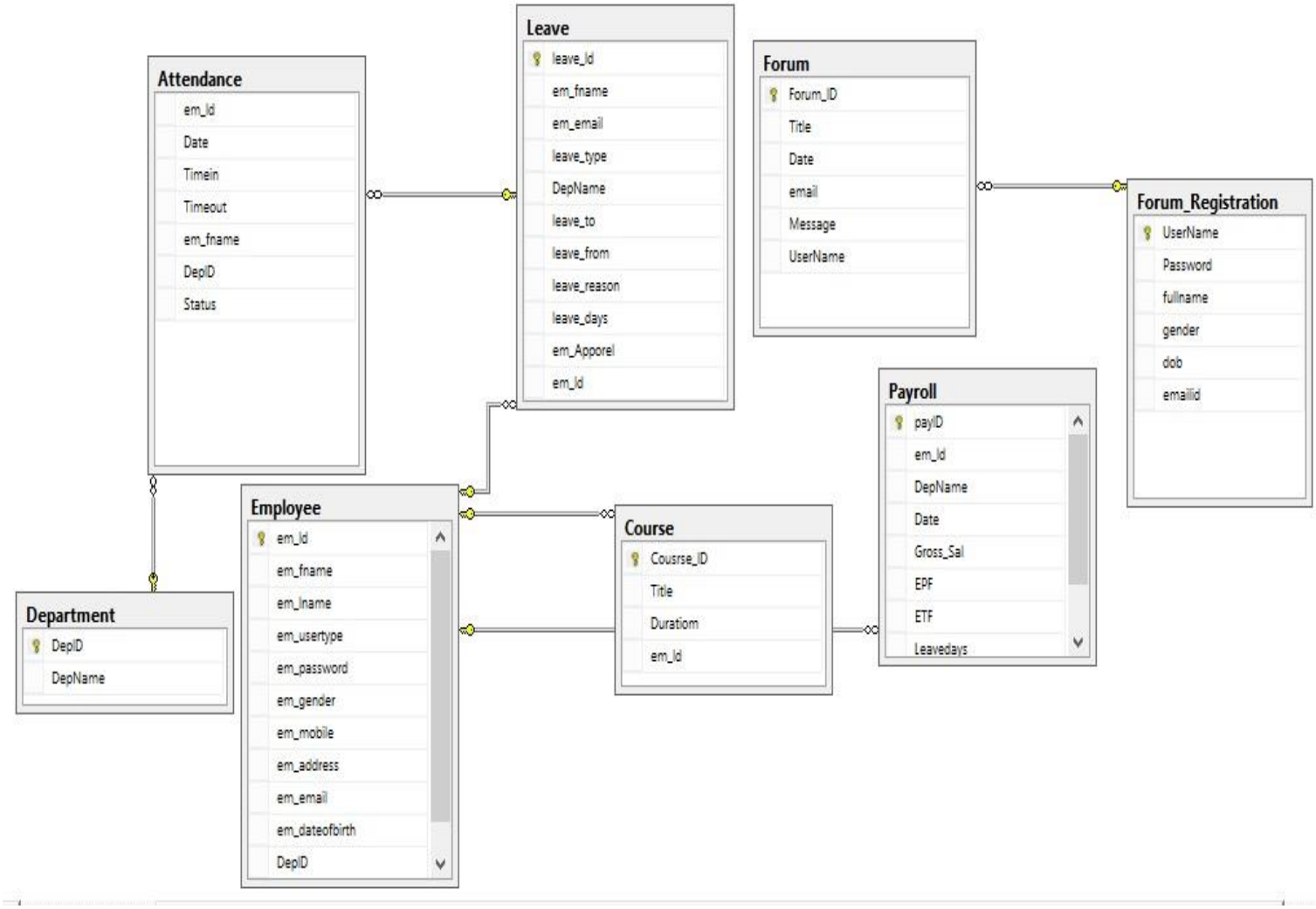


Table Views

Employee

HOME.HR - dbo.Employee

em_Id	em_fname	em_lname	em_usertype	em_password	em_gender	em_mobile	em_address	em_email	em_dateofbirth	DepID
1	Kasun	Thivanka	Admin	12345	Male	0713482071	No.33,Minuwa...	kasunthivanka...	1993-03-14	D1001
2	Malith	test 1	Emplyee	123	Male	456465	cdscs	cscds@ghgvhj....	2161-01-01	D1001
3	Milinda	Kaushika	HR	12345	Male	0713482081	No.31,Minuwa...	milindakaushik...	1992-03-15	D1001
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Attendance

HOME.HR - dbo.Attendance

em_Id	Date	Timein	Timeout	em_fname	DepID	Status
1	2018-03-16 00:0...	2018-03-16 00:0...	2018-03-17 05:0...	Kasun	IT Department	Present
2	2018-03-27 00:0...	2018-03-27 08:0...	2018-03-16 05:0...	Milinda	Admin Depart...	Absent
3	2018-03-16 00:0...	NULL	NULL	Salika	Marketing Deve...	Absent
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Forum

HOME.HR - dbo.Forum

Forum_ID	Title	Date	email	Message	UserName
1	What is HTML5?	2018-04-03	Salika@gmail.c...	HTML5 is a cor...	Salika
5	What is WWW	2018-04-05	Salika123@gma...	The Web, or W...	Salika
14	What is string?	2017-02-05	Prashant@gma...	Strings. Nearly ...	Prashant
NULL	NULL	NULL	NULL	NULL	NULL

Forum registration

HOME.HR - dbo.Forum_Registration

UserName	Password	fullname	gender	dob	emailid
Prashant	12345	PrashantSilva	Male	1993-08-21	Prashant@gma...
Salika	E1$$MCM	SalikaAgunwala	Male	2018-04-04	salikaagunwala...
NULL	NULL	NULL	NULL	NULL	NULL

Course

HOME.HR - dbo.Course

Cousrse_ID	Title	Duratiom	em_Id
1001	Computing	good	1
NULL	NULL	NULL	NULL

Department

HOME.HR - dbo.Department

	DepID	DepName
▶	D1001	Admin Depart...
	D1002	Account Depart...
	D1003	Marketing Dep...
	D1004	IT Department
	D1006	Production Dep...
	D1007	HR Development
*	NULL	NULL

Salary Calculation

HOME.HR - dbo.Payroll

	payID	em_Id	DepName	Date	Gross_Sal	EPF	ETF	Leavedays	Net_Sal
▶	P10001	3	IT Development	2018-04-03	25000.00	5000.00	250.00	6	17250.00
	P10002	3	IT Development	2018-04-02	20000.00	2000.00	750.00	6	13800.00
	P10003	1	Admin Depart...	2018-04-02	20000.00	2000.00	1500.00	2	16466.67
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Leave Info

HOME.HR - dbo.Leave

	leave_Id	em_fname	em_email	leave_type	DepName	leave_to	leave_from	leave_reason	leave_days	em_Apporel	em_Id
▶	1	Kasun	kasun@gmail.c...	FullDay	Admin Depart...	2018-03-23	2018-03-17	sick	2	Approvel	1
	2	Malith	malith@gmail....	FullDay	IT Development	2018-03-17	2018-03-27	Sick	6	Waiting for Ap...	3
	3	Milinda	milindakaushik...	Halfday	Account Depart...	1900-01-01	1900-01-01	Sick	1.5	Waiting for Ap...	2
	12	Milinda	infogmail.com	FullDay	Account Depart...	2018-03-17	2018-03-16	Sick	2	Waiting for Ap...	2
	15	Milinda	infogmail.com	FullDay	Account Depart...	2018-03-23	2018-03-22	Sick	1	Reject	2
	16	Milinda	infogmail.com	FullDay	Account Depart...	2018-03-23	2018-03-22	Sick	1	Approvel	2
	18	Kasun	milindakaushik...	Halfday	Admin Depart...	1900-01-01	1900-01-01	Sick	1.5	Waiting for Ap...	1
	19	Kasun	milindakaushik...	FullDay	Admin Depart...	2018-03-29	2018-03-16	Sick	13	Waiting for Ap...	1
	20	Milinda	milindakaushik...	FullDay	Admin Depart...	2018-04-05	2018-04-13	Sick	-8	Waiting for Ap...	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Own Reflections

1. Milinda Kaushika

First of all, I thank you for selecting me as a professional Object-Oriented Programming developer to design and implement an HR Management System.

Though there are many programming languages in the world, Asp.net Programming language is selected to design this HR Management System. The reasons for selecting Asp.Net are given below. Asp.Net is the foundation for virtually every type of networked application. Asp.Net has been tested, refined, extended, Asp.Net applications are used across heterogeneous environments, businesses can provide more services and boost end-user productivity and communication, as a result it reduces the cost of ownership of both enterprise and consumer applications.

Visual Studio C#, ASP.NET

• It is easy to handle event in Visual Studio C#, ASP.NET when compare to other Development tools. Because in PHP we have to use scripting language (JavaScript) to capture and handle the event but ASP.NET allows to write events in code behind.

• Microsoft ASP.NET contains special AJAX controls which can be used to create AJAX enabled web applications very easily. If we use PHP then we have to use scripting language to write AJAX.

• Buttons has command Argument function therefor parameters can pass without keeping hidden fields.

To obtain the mentioned advantages I recommend to use ASP.NET for this solution to achieve efficiency, productivity and communication, as a result it reduces the cost of ownership. I have less experience in ASP.NET. With the support and the features provided by the Visual Studio IDE and SQL Server Management Studio I was able to learn and develop the system with my team member successfully.

I have stated with definition of .NET Reflection, list of mostly used classes the System. Reflection namespace provides and importance of Type class in .net and methods. I was use of properties and methods of Type class for the .NET web application.

2. Mohamed Arham Khan

This was my very first ASP.NET project in my career and I have gained a lot thorough this.

I had a very good chance on getting experience in .NET programming through this course work. Also, I had a very good chance to work with web applications and database Server too.

I had to refer many tutorials on internet that polished my knowledge on this.

I am very happy to sign of this project by gaining more knowledge in software development and object-oriented concepts. I look forward to take my knowledge to the next step from where I am now.

3. K. Aroonpragash.

   I had a very good chance on getting experience in C# and ASP. NET programming through this course work. Also, I had a very good chance to work with MS SQL Server database too.

   As I didn't know ASP.NET before, it is completely new to me. I had to refer tutorials available on the Internet. Also, my group members assisted in me in all possible ways in order to complete my part in this assignment.

   ASP.NET allows to write less lines of codes to larger problems, allows to combine the knowledge of HTML, it shows clear alerts on the memory leaks to delete all memory areas not needed further more, unbound loops, etc.

   I am so happy at the end of this assignment, that I have learnt a quite a lot of ASP.NET and obtained a lot of knowledge in working with C# along with object-oriented concepts.

Project Plan

	Week 1							Week 2							Week 3							Week 4						
Section	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Initial Study																												
System Design																												
Development																												
Testing																												
Documentation																												

Testing

System testing is designed to uncover the weaknesses that were not founding earlier test. In the testing phase, the program is executed with the explicit intention of finding errors. This includes forced system failures and validation of the system, as its user in the operational environment will implement it. For this purpose test cases are developed. When a new system replaces the old one, such as in the present case, the organization can extract data from the old system to test them on the new. Such data usually exist in sufficient volume to provide sample listings and they can create a realistic environment that ensures eventual system success. Regardless of the source of test data, the programmers and analyst will eventually conduct four different types of tests.

Integration Testing

The integration is the next important concept that highlights in the testing scenario. Integration testing can be performed in different strategies. One of them is the Big Bang testing in which one could first test all of a system's modules separately and then whole systems at once. But here we proceed abruptly from the module testing and the integration testing disappears. Another alternative is the Incremental Testing. With the Incremental testing there are many advantages. We can start the integration as soon as reasonable subsets of modules have been developed. It is easier to localize errors incrementally. The partial aggressions of modules often constitute important subsystems that can have autonomy with these testing. The need for stubs and drivers can be reduced.

There are two approaches to the Incremental Testing. They include Bottom-up and Top-down aggregations. The former means starting aggregation and testing from leaves of the module charts. The latter means starting from the top-level modules and substitute for higher-level modules. In our project we have used the top-down approach of incremental testing. Top-down integration is an incremental approach to the construction of programs structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module that is the basic connectivity module in our project. Test is done on each module. The top down integration strategy verifies major control or decision points. In the beginning of the integration phase dummy frames were selected as stubs to ensure that the data flow occurred through the correct hierarchical structure. Later the actual module replaces these stubs.

Various Testing Methods

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as Module Testing. The modules are tested separately. This testing is carried out during programming stage itself.

Validation Testing

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the users .After validation test has been conducted one of the two possible conditions exists1. The function or the performance characteristics confirm to specification and are accepted.2. A deviation from specification is uncovered and a deficiency lists created.

Output Testing

After performing the validation testing the next step is output testing of the proposed system since no system is useful if it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested by asking the users about the formats required by them.

Quality Assurance Methodologies

Quality assurance is a planned and systematic of all actions necessary to provide adequate confidence that the item or product confirms to established technical requirements. The purpose of software quality assurance group is to provide assurances that the procedures, tools and techniques used during product development and modification and adequate to provide desired level of confidence in the work products. Often, software quality assurance personnel are organizationally distinct from software development group. Preparation of a Software Quality Assurance

Plan for each software products is primary responsibility of software quality assurance group.

System verification and validation

System verification and validation is done to check the quality of the software in simulated and live environment. A number of different transactions are used to perform verification. Validation is the process of demonstrating that the implemented software does satisfy the system requirements. One aspect of software validation is to statistically analyze the program without resorting tactual execution. The system validation done in such-a-way that the system response time will not cause any hardship to the user.

White Box Testing

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, we can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structures to ensure their validity.

Black Box Testing

Black box testing methods focus on the functional requirements if the software. That is, black box testing enables us to derive sets of input conditions that will fully exercise all functional requirements of the program. Black box testing attempts to find errors in following categories:

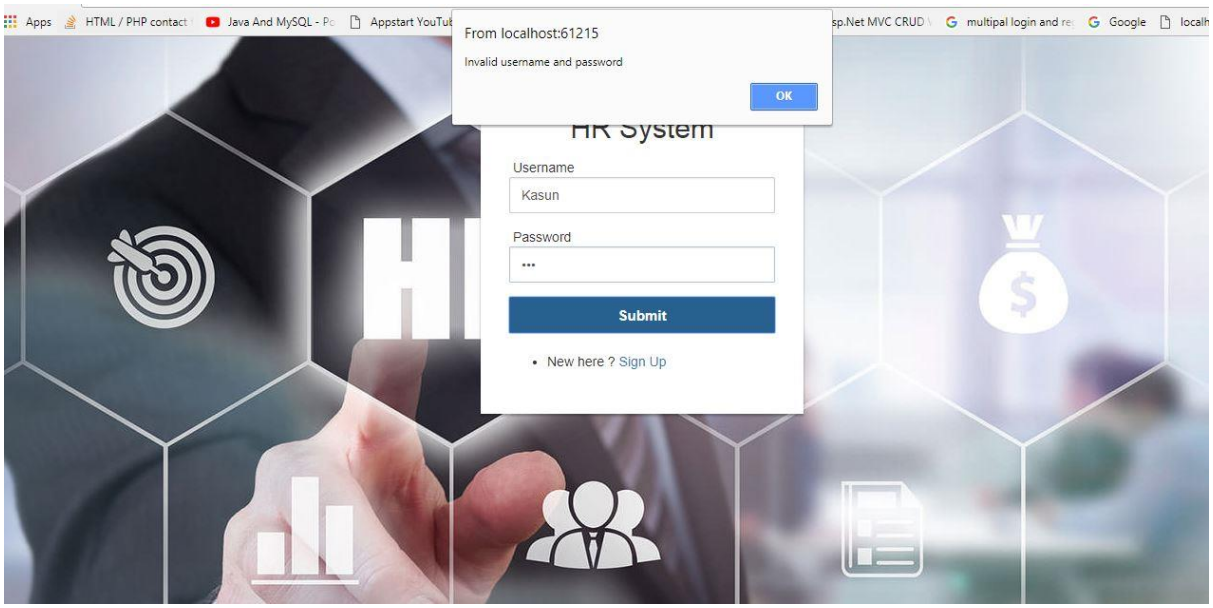
- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database.
- Performance errors Initialization and termination errors.

Text plan execution and test results

Test case 1 execution

- **Test case 1 – Login Information.**

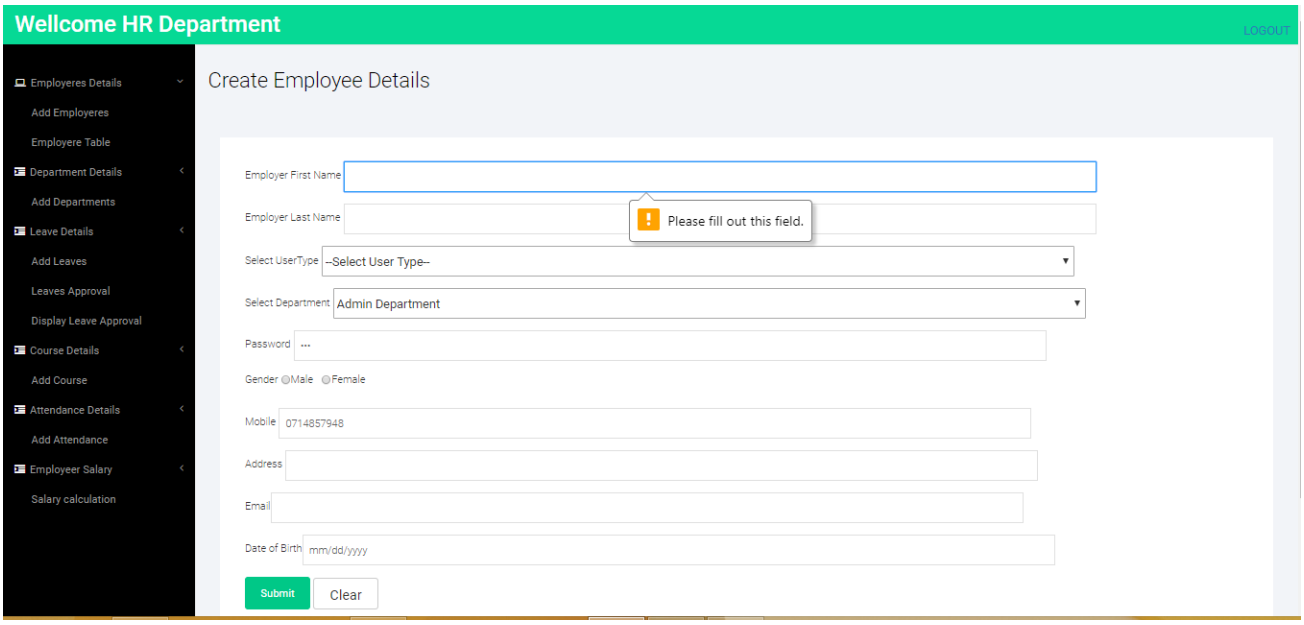
Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0001	Login to the system using invalid credentials	1. Enter "milinda1" to the Username 2. Enter "1235" to the Password 3. Press "Login" button	system should give an error saying input username or password is wrong	F	2-April-2018	M	
TC0002	Login to the system using valid credentials	1. Enter "milinda" to the Username 2. Enter "12345" to the Password 3. Press "Login" button	Successful login for Dashboard.	P	2-April-2018	M	

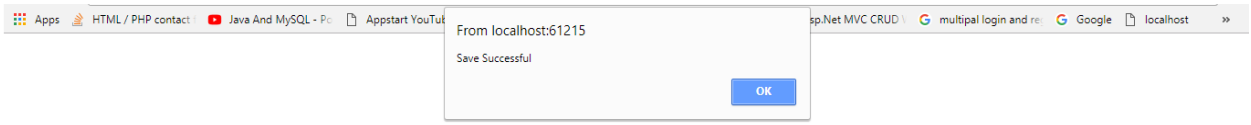


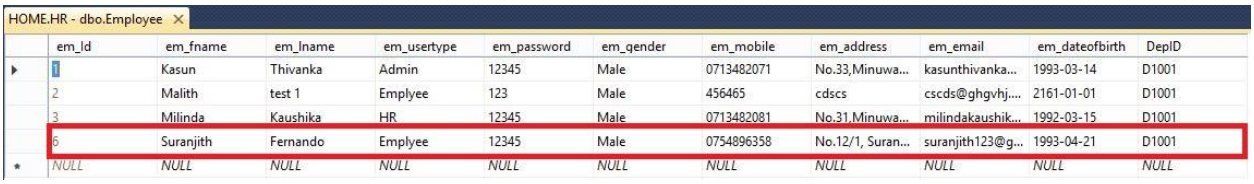
Test case 2 execution

- **Test case 2 – Employee Information.**

Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0003	Try to save the employee form without entering any field	1. Go to Employee Registration form 2. Press Save button	System should give an warning or it should show the missing fields.	F	2-April-2018	M	
TC0004	Save all required fields and cross check with database	1. Save Employee. 2. Registration form. 3. Fill out all the required fields. 4. Press Save.	System should save all required fields in the database.	P	2-April-2018	M	





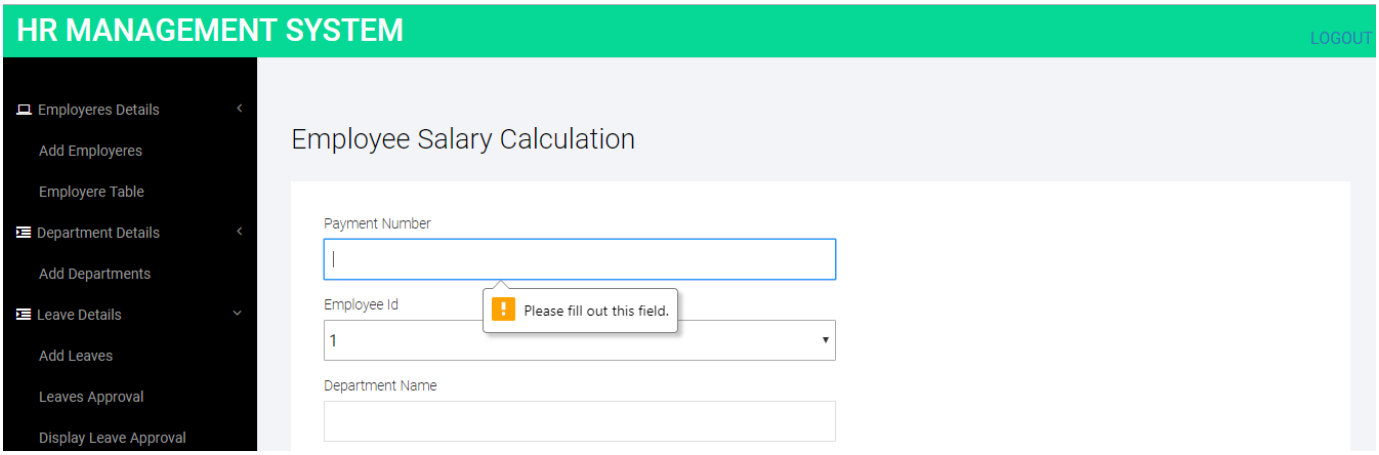


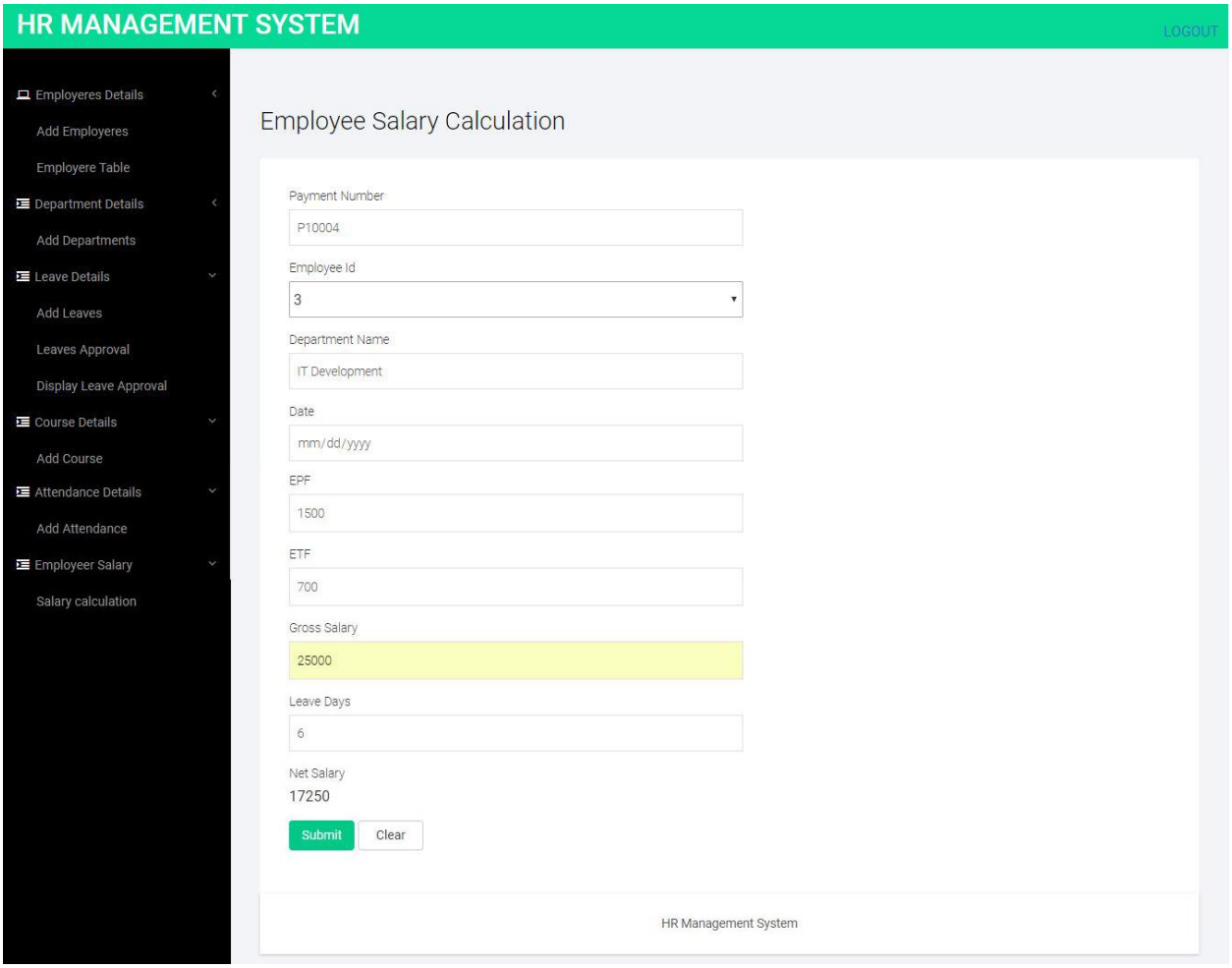
em_Id	em_fname	em_lname	em_usertype	em_password	em_gender	em_mobile	em_address	em_email	em_dateofbirth	DepID
1	Kasun	Thivanka	Admin	12345	Male	0713482071	No.33,Minuwa...	kasunthivanka...	1993-03-14	D1001
2	Malith	test 1	Emplyee	123	Male	456465	cdscs	cscds@ghgvhj....	2161-01-01	D1001
3	Milinda	Kaushika	HR	12345	Male	0713482081	No.31,Minuwa...	milindakaushik...	1992-03-15	D1001
6	Suranjith	Fernando	Emplyee	12345	Male	0754896358	No.12/1, Suran...	suranjith123@g...	1993-04-21	D1001
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Test case 3 execution

- **Test case 3 – Employee Payment Information.**

Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0005	Try to save the Payment details without entering any field	1. Go to Employee Salary Calculation Form 2. Press Save button	System should give an warning or it should show the missing fields.	F	2-April-2018	M	
TC0006	Try to save the without EFF,ETF, Gross salary and Leave days.	1. Fill out the required fields. 2. Press Save.	System should give an warning or it should show the missing fields.	F	2-April-2018	M	
TC0007	Save all required fields and check with database	1. Fill out all the required fields. 2. Press Save.	System should save all required fields in the database.	P	2-April-2018	M	





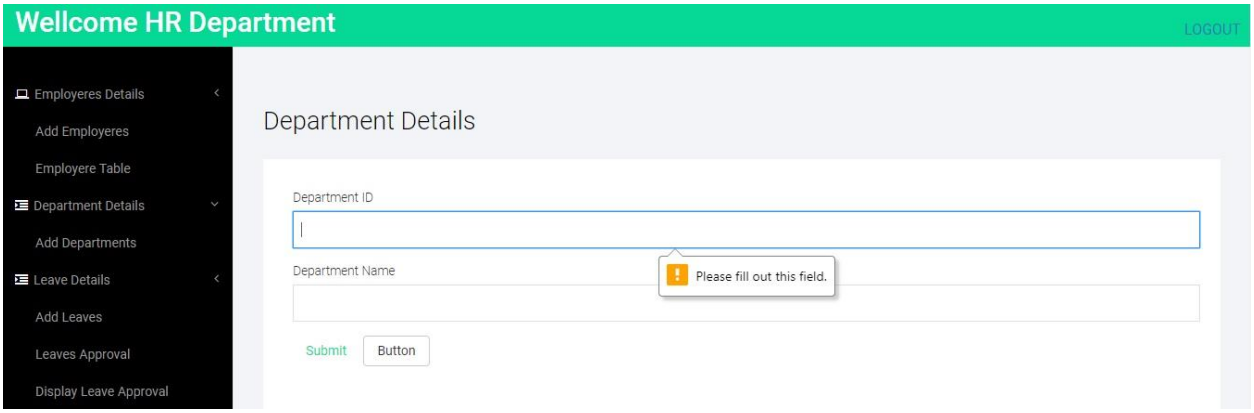
HOME.HR - dbo.Payroll

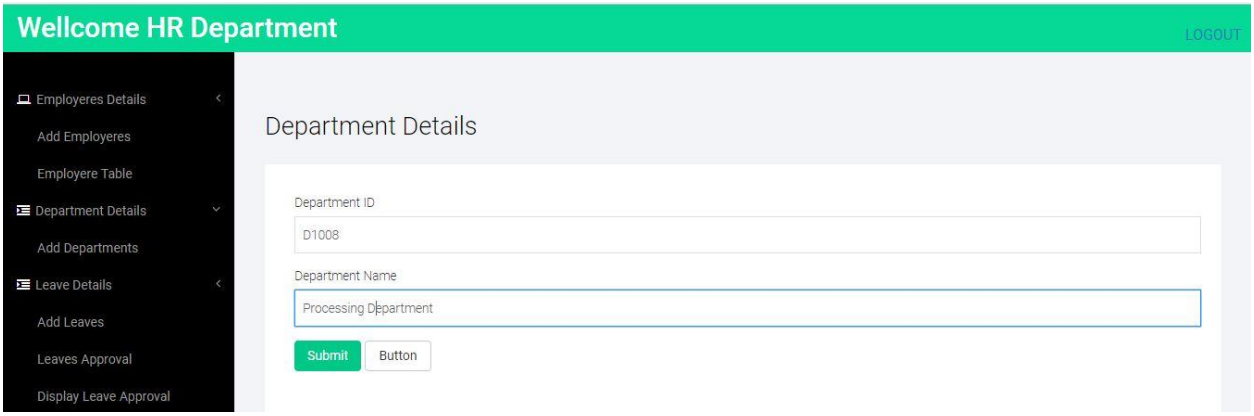
payID	em_Id	DepName	Date	Gross_Sal	EPF	ETF	Leavedays	Net_Sal
P10001	3	IT Development	2018-04-03	25000.00	5000.00	250.00	6	17250.00
P10002	3	IT Development	2018-04-02	20000.00	2000.00	750.00	6	13800.00
P10003	1	Admin Depart...	2018-04-02	20000.00	2000.00	1500.00	2	16466.67
P10004	3	IT Development	2018-04-05	25000.00	1500.00	700.00	6	17250.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

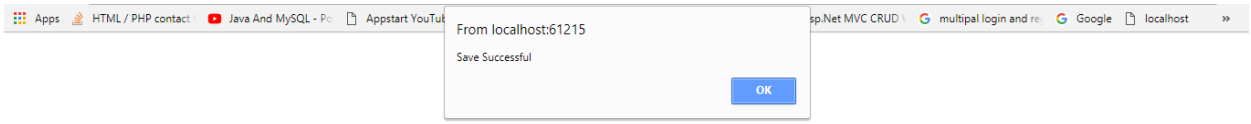
Test case 4 execution

- **Test case 4 – Department Details.**

Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0008	Try to save the department details without entering any field.	1. Go to Department Form. 2. Press Save button	System should give an warning or it should show the missing fields.	F	5-April-2018	M	
TC0009	Save all required fields and cross check with database	2. Fill out all the required fields. 3. Press Save.	System should save all required fields in the database.	P	5-April-2018	M	







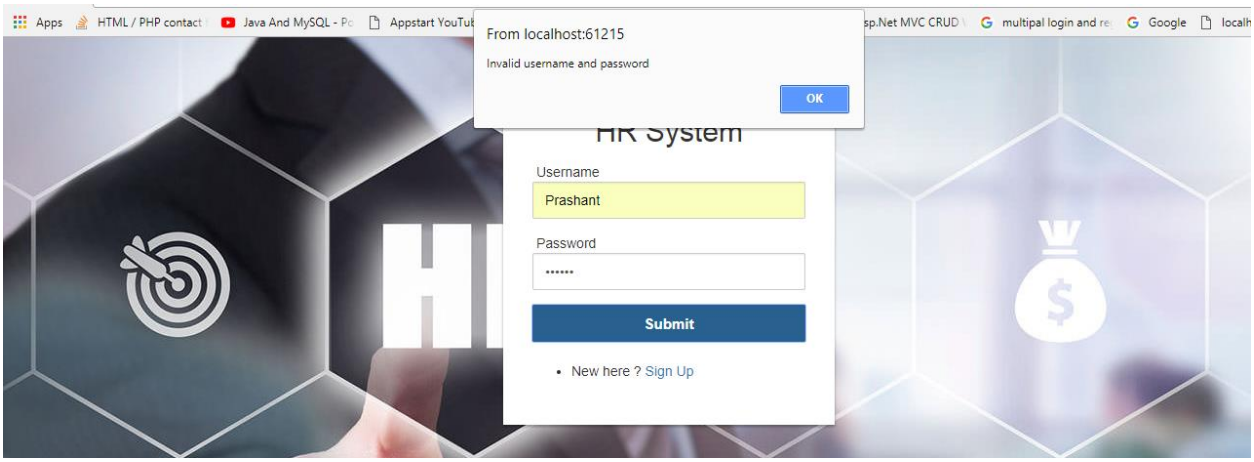
HOME.HR - dbo.Department

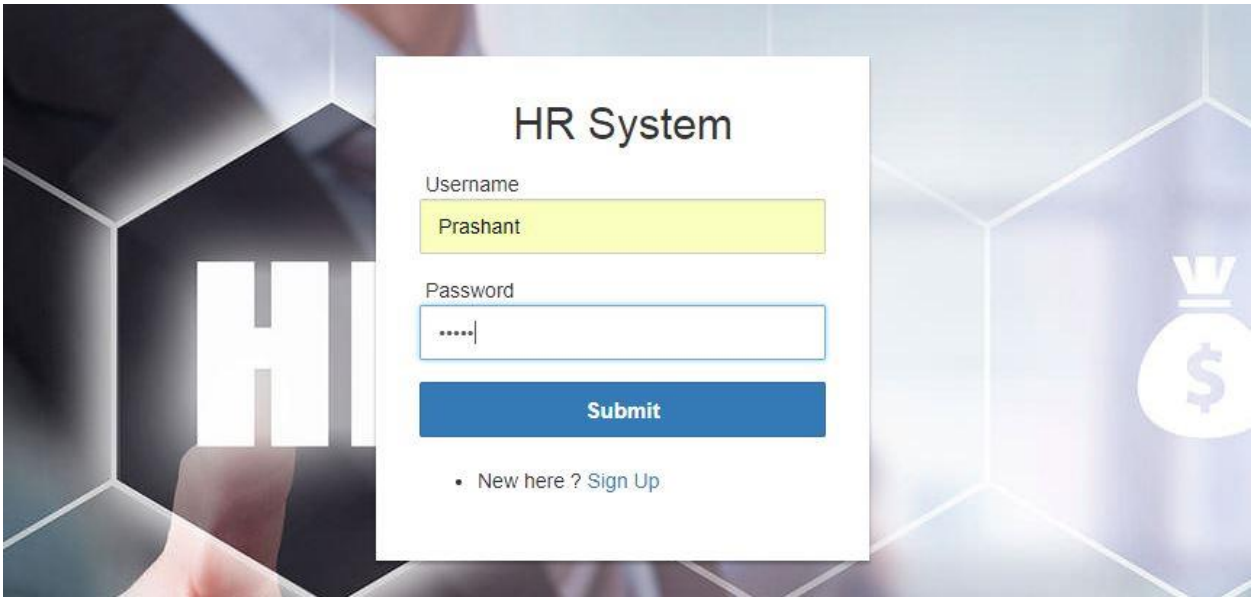
DepID	DepName
D1001	Admin Depart...
D1002	Account Depart...
D1003	Marketing Dep...
D1004	IT Department
D1006	Production Dep...
D1007	HR Development
D1008	Processing Dep...
NULL	NULL

Test case 5 execution

- **Test case 5 – Feedback Login Details.**

Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0010	Login to the system using invalid credentials	1. Enter " Prashant123" to the Username 2. Enter " 1234" to the Password 3. Press "Login" button	system should give an error saying input username or password is wrong	F	2-April-2018	M	
TC0011	Login to the system using valid credentials	1. Enter " Prashant " to the Username 2. Enter "12345" to the Password 3. Press "Login" button	Successful login for Dashboard.	P	2-April-2018	M	

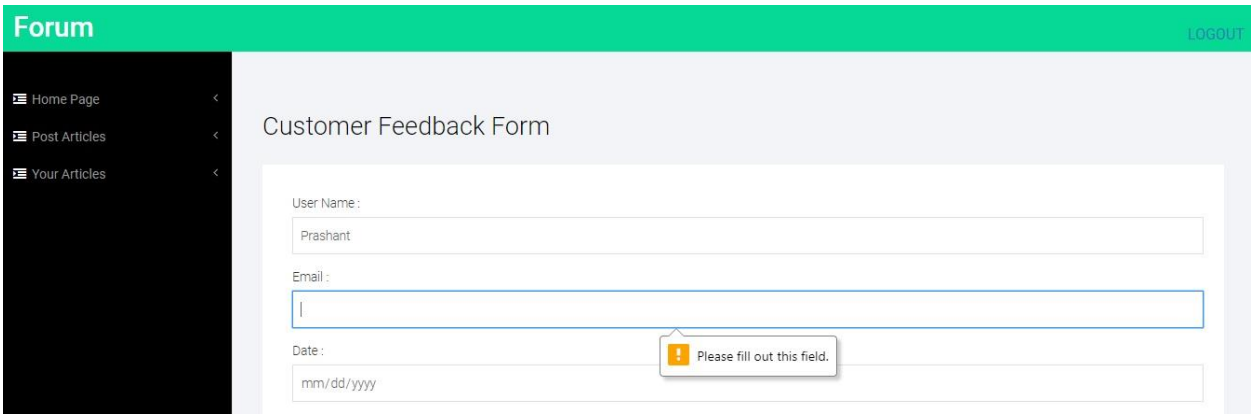


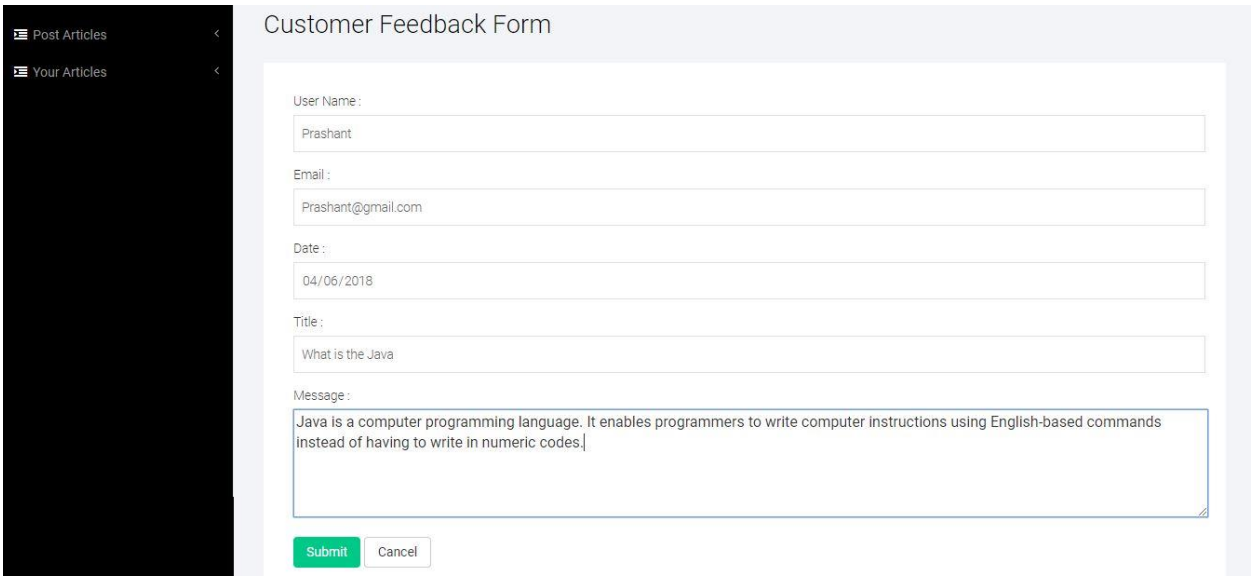


Test case 6 execution

- **Test case 6 – Feedback form Details.**

Test Case #	Condition	Test Execution step	Expected Result	Test Result	Date Tested	Tester	Comments
TC0010	Try to save the feedback form without entering any field.	2. Go to Feedback Form. 2. Press Save button	System should give an warning or it should show the missing fields.	F	5-April-2018	M	
TC0011	Save all required fields and cross check with database	1. Fill out all the required fields. 2. Press Save.	System should save all required fields in the database.	P	5-April-2018	M	





HOME.HR - dbo.Forum

Forum_ID	Title	Date	email	Message	UserName
1	What is HTML5?	2018-04-03	Salika@gmail.c...	HTML5 is a cor...	Salika
5	What is WWW	2018-04-05	Salika123@gma...	The Web, or W...	Salika
14	What is string?	2017-02-05	Prashant@gma...	Strings. Nearly ...	Prashant
15	What is the Java	2018-04-06	Prashant@gma...	Java is a compu...	Prashant
NULL	NULL	NULL	NULL	NULL	NULL

References

What is the object-oriented programming http://searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP/[ONLINE] [Accessed 05 April 2018].

object-oriented programming Objects, Classes & Methods http://study.com/academy/lesson/oop-object-oriented-programming-objects-classes-interfaces.html/[ONLINE] [Accessed 03 April 2018].

http://searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP/[ONLINE] [Accessed 03 April 2018].

There are four types of object-oriented programming concepts. http://codebetter.com/raymondlewallen/2005/07/19/4-major-principles-of-object-oriented-programming/ [ONLINE] [Accessed 30 March 2018].