(11)

PROJECT REPORT
ON
EMPLOYEE PROFILE MANAGEMENT
USING
ASP.NET

SUMMER INTERNSHIP 2016

ATTENDED AT

NIH ROORKEE

SUBMITTED BY:

SAHIL MADAN

For the partial fulfilment of the

Degree of B.Tech in INFORMATION TECHNOLOGY

USICT, GGSIPU
NEW DELHI

CERTIFICATE

We hereby certify that the project work entitled "Employee Profile Management" carried out by Mr. Sahil Madan, B.Tech (Information Technology), USICT, GGSIPU, New Delhi is authentic record of work carried out by him during June 02, 2016 to August 02, 2016 under our guidance.

We appreciate the sincerity and hard work put by Mr. Sahil Madan for completing the task in 60 days period. We wish him success in his future endeavour.

(Dr. A.K Lohani)

Scientist 'G' & Head
Computer Centre,
NIH Roorkee

(Abhishek Agarwal)

Resource Person (Junior)
Computer Centre,
NIH Roorkee

ACKNOWLEDGEMENT

I would like to express my profound gratitude and thanks to respected Dr. A.K. LOHANI Scientist 'G', for his immense support, valuable guidance and advices during the preparation of this project. He helped nurse this project from idea of stage to the record. I tried my maximum to make this project according to his suggestion and direction.

I am also thankful to Sri Manoj Goel, SRA who never hesitated to extend his helping hand whenever required during the project work. I also extend my sincere thanks to Mr. Abhishek Agarwal, Resource Person (Junior) for support and also making available the computer resources. I want to acknowledge all my teachers and colleagues for their continuous help and support.

By: Sahil Madan

B.Tech(IT)

ABSTRACT

Nowadays, with the evolution of the Internet and the World Wide Web, there is a need for websites and web applications, and with the help of .NET framework we can create these. This report consists of the basic knowledge about .NET framework and a project titled "Employee Profile Management". The project is a Human Resource management application that delivers effective employee data management and integrated directory services required for the management of database of the employees.

The project simply aims to maintain the details provided by the employees in the database. For this purpose, the softwares used are "Microsoft SQL Server 2008" for backend and "Microsoft Visual Studio 2012" for frontend. The languages used in the project are "C#" and "asp.net", Various features of Visual Studio and asp.net are used in combination here like Sessions, master pages, etc.

In the project the employee can log into his account and enter or change his personal or professional details. The user needs to create a password and in case he forgets he also has an option to change it. Also, the security measures have been taken into place by storing the password in database in an encrypted form. Hence, this system will enhance the functioning of maintenance of the details of employees.

INDEX

6.	Employee Profile Management -MasterPage
7.	GUI and Code -Sign In -Password and Security Question -Log In -Forgot Password -New Password -Employee Details -About Us -Contact Us
8.	References

About NIH

NATIONAL INSTITUTE OF HYDROLOGY, the premier Institute in the area of hydrology and water resources in India. The Institute was established in 1978 with the main objective of undertaking, aiding, promoting and coordinating systematic and scientific work in all aspects of hydrology. The Institute has its Headquarters at Roorkee (Uttarakhand), four regional centres at Belgaum, Jammu, Kakinada and Bhopal and two centres for Flood Management Studies at Guwahati and Patna. The Institute is well equipped to carry out computer, laboratory & field oriented studies.

THE ROLE AND FUNCTIONS OF NATIONAL INSTITUTE OF HYDROLOGY

- To undertake, aid, promote and coordinate systematic and scientific work on all aspects of hydrology.
- To cooperate and collaborate with other national, foreign and international organizations in the field of hydrology.
- To establish and maintain a research reference library in pursuance of the objectives of the Society and equip the same with books, reviews, magazines and other relevant publications.
- To do all other such things as the Society may consider necessary, incidental or conducive for the attainment of the objectives for which the Institute has been established.

AREAS OF SPECIALIZATION:-

- ENVIRONMENTAL GROUND
- WATER
- HYDROLOGY
- HYDROLOGY
- SURFACE WATER
- WATER
- RESOURCES
- HYDROLOGY
- SYSTEMS
- HYDROLOGICAL
- INVESTIGATIONS
- ACTIVITIES

BASIC, APPLIED AND STRATEGIES

- RESEARCH
- SOFTWARE DEVELOPMENT
- USER DEFINED, DEMAND DRIVEN RESEARCH PROJECTS

- CONSULTANCY
- TECHNOLOGY TRANSFER
- INCOH SECRETARIAT
- WATER AVAILABILITY

Computer Centre in NIH ROORKEE:-

The Computer Centre is responsible for providing computing, networking, internet and email facilities to Scientists and Staff at the Institute. Computing has become an essential tool for almost all scientific research. Computers of latest configuration are available in the Institute for conducting complex hydrological analyses and modelling studies. Internet and email facilities are routinely utilized for accessing scientific literature as well as for correspondence. The Institute also uses computers for processing of administrative and accounting data. The centre procures and provides maintenance to computers and related peripherals available with the Scientists and Staff.

The local area network (LAN) provides interconnectivity between the computers in different building blocks of the campus. The network comprises of switches in various blocks and UTP connections to individual machines. The LAN is connected to Internet by a 50 Mbps leased line link from BSNL.VSAT connectivity from ERNET India also exists for email communication. A centralized server receives and

Stores/forwards emails to respective users. A dedicated web server (www.nih.ernet.in) provides a platform for hosting institute information, research publications, important announcements, tender notices etc. and some useful hydrology related information. To maintain the integrity and security of Institute's network, a firewall

restricts access from outside to machines within the LAN. All the facilities existing at the Computer Centre are upgraded from time to time to meet the evolving standards of Information Technology.

INTRODUCTION TO .NET

.Net is a set of technologies provided by Microsoft that facilitates application development by offering multiple languages well suited for different tasks as well as preferences, an integrated development environment for productive application development, a class library providing core functionality to all applications, and an execution environment for managed execution of the code on any platform. The database used in .Net is SQL Server and the server used is Internet Information Server (IIS).

Execution of .Net Applications

Common Language Runtime (CLR) is the application where a .Net application executes. It acts as a platform for all .Net applications i.e. it is responsible for providing execution environment to applications, allocating memory, garbage collection and implementing type safety.

.Net is a **Platform Independent** technology. Platform Independence is one of the most frequently used as well as least understood terms. Commonly believed definition of platform independence is " A language is platform independent if a program written in this language executes on all platforms without any modification in the source code of the program." The role of platform in .Net is played by Common Language Runtime, which is a part of .Net framework.

Components of .Net

Languages

VB.Net

C#.Net

ASP.Net, others

These act as tools.

Class Libraries act as model.

Common Language Runtime (CLR) act as Platform.

Features of .Net

Following are some of the salient features of .Net:

- **Platform Independence :** .Net is a **Platform Independent** technology. Platform Independence is one of the most frequently used as well as least understood terms. Commonly believed definition of platform independence is " A language is platform independent if a program written in this language executes on all platforms without any modification in the source code of the program."

- **Language Independence :** Language Independence means that the development of an application doesn't depend on any single language, i.e. different modules of an application can be completed in different languages which can be seamlessly interfaced. .Net supports various languages like VB.Net, C#.Net, ASP.Net, etc. to suit the preferences of different users as well as to provide fine tuned performance from different languages for specialized task. Apart from this Microsoft has issued Common Language Specification (CLS) to allow the entry of new languages within the sphere of .Net.

- **XCopy Deployment :** One of the major changes brought in by .Net is the simplification of deploying or installing an application. In .Net the basic unit of deployment is an assembly that is a self describing deployment package. It contains MSIL code of application classes, and XML files that describe types, referenced libraries and other information required by a run

time environment for successful execution of an application. An application can be deployed by simply copying its assembly to the target machine.

- **Managed Code Execution :** Managed Code Execution means that the programmer is not bothered by pesky tasks such as finding the availability of memory, allocation or de-allocation of memory during the execution of the program. All these tasks are efficiently performed by the run-time environment in the case of .Net CLR.

Types of Applications supported by .Net:

Following types of applications can be developed in .Net:

1. Console Application
2. Windows Application
3. ASP.Net Web Application
4. Class Library
5. ASP.Net Web Service
6. Windows Control Library
7. Smart Device Application
8. ASP.Net Mobile Web Application
9. Web Control Library

.Net IDE (Integrated Development Environment) is Microsoft Visual Studio and the database used is SQL Server.

INTRODUCTION TO C#

C# (pronounced as C sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg.

C# program:

To create and run a console application

1. Start Visual Studio.
2. On the menu bar, choose **File**, **New**, **Project**.
   The **New Project** dialog box opens.
3. Expand **Installed**, expand **Templates**, expand **Visual C#**, and then choose **Console Application**.
4. In the **Name** box, specify a name for your project, and then choose the **OK** button.
   The new project appears in **Solution Explorer**.
5. If Program.cs isn't open in the **Code Editor**, open the shortcut menu for **Program.cs** in **Solution Explorer**, and then choose **View Code**.
6. Replace the contents of Program.cs with the following code.

```
// A Hello World! program in C#.
using System;
namespace HelloWorld
{
  class Hello
  {
    static void Main()
    {
      Console.WriteLine("Hello World!");

      // Keep the console window open in debug mode.
      Console.WriteLine("Press any key to exit.");
      Console.ReadKey();
```

```
    }
  }
}
```

7. Choose the F5 key to run the project. A Command Prompt window appears that contains the line Hello World!

Data Types:

The varibles in C#, are categorized into the following types:

- **Value types** : To get the exact size of a type or a variable on a particular platform, you can use the **sizeof** method. The expression *sizeof(type)* yields the storage size of the object or type in bytes.
- **Reference types**: The reference types do not contain the actual data stored in a variable, but they contain a reference to the variables.

  In other words, they refer to a memory location. Using multiple variables, the reference types can refer to a memory location. If the data in the memory location is changed by one of the variables, the other variable automatically reflects this change in value.

- **Pointer types** : Pointer type variables store the memory address of another type. Pointers in C# have the same capabilities as the pointers in C or C++. Syntax for declaring a pointer type is

  type* identifier.

Inheritance: One of the most important concepts in object-oriented programming is inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and speeds up implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base**class, and the new class is referred to as the **derived** class.

The idea of inheritance implements the **IS-A** relationship. For example, mammal **IS A** animal, dog **IS-A** mammal hence dog **IS-A** animal as well, and so on.

Types of Inheritance:

- Single Inheritance
- Multilevel Inheritance
- Hybrid Inheritance
- Hierarchical Inheritance

Polymorphism: The word **polymorphism** means having many forms. In object-oriented programming paradigm, polymorphism is often expressed as 'one interface, multiple functions'.

Polymorphism can be static or dynamic. In **static polymorphism**, the response to a function is determined at the compile time. In **dynamic polymorphism**, it is decided at run-time.

Interfaces: An interface is defined as a syntactical contract that all the classes inheriting the interface should follow. The interface defines the **'what'** part of the syntactical contract and the deriving classes define the **'how'** part of the syntactical contract.

Interfaces define properties, methods, and events, which are the members of the interface. Interfaces contain only the declaration of the members. It is the responsibility of the deriving class to define the members. It often helps in providing a standard structure that the deriving classes would follow.

Abstract classes to some extent serve the same purpose, however, they are mostly used when only few methods are to be declared by the base class and the deriving class implements the functionalities.

Delegates: C# delegates are similar to pointers to functions, in C or C++. A **delegate** is a reference type variable that holds the reference to a method. The reference can be changed at runtime.

Delegates are especially used for implementing events and the call-back methods. All delegates are implicitly derived from the **System.Delegate** class.

Collections: Collection classes are specialized classes for data storage and retrieval. These classes provide support for stacks, queues, lists, and hash tables. Most collection classes implement the same interfaces.

Collection classes serve various purposes, such as allocating memory dynamically to elements and accessing a list of items on the basis of an index etc. These classes create collections of objects of the Object class, which is the base class for all data types in C#.

ADO.NET (ActiveX Data Object)

ADO.NET

Most applications need data access at one point of time making it a crucial component when working with applications. Data access is making the application interact with a database, where all the data is stored. Different applications have different requirements for database access. C#.NET uses ADO.NET(ActiveX Data Object) as it's data access and manipulation protocol which also enables us to work with data on the Internet.

Evolution of ADO.NET

The first data access model, DAO(data access model) was created for local databases with the built in Jet engine which had performance and functionality issues. Next came RDO(Remote Data Object) and ADO(Active Data Object) which were designed for Client Server architectures but soon ADO took over RDO. ADO was a good architecture but as the language changes so is the technology. With ADO, all the data is contained in a record set object which had problems when implemented on the network and penetrating firewalls. ADO was a connected data access, which means that when a connection to the database is established the connection remains open until the application is closed. Leaving the connection open for the lifetime of the application raises concerns about database security and network traffic.

Importance of ADO.NET

ADO.NET maintains a disconnected database access model which means, when an application interacts with the database, the connection is opened to serve the request of the application and is

closed as soon as the request is completed. Likewise, if a database is updated, the connection is opened long enough to complete the Update operation and is closed. By keeping connections open for only a minimum period of time, ADO.NET conserves system resources and provides maximum security for databases and also has less impact on system performance. Also, ADO.NET when interacting with the database uses XML and converts all the data into XML format for database related operations making them more efficient.

The ADO.NET Data Architecture

Data Access in ADO.NET relies on two components: DataSet and Data Provider.

DataSet: The DataSet is disconnected, in memory representation of data. It can be considered as a local copy of the relevant portions of the database. The DataSet is persisted in memory and the data in it can be manipulated and updated independent of the database.

Data Provider: The Data Provider is responsible for providing and maintaining the connection to the database. A Data Provider is a set of related components that work together to provide data in an efficient and performance driven manner. The .NET Framework currently comes with two Data Providers: the SQL Data Provider which is designed only to work with Microsoft's SQL server 7.0 or later and the OleDb Data Provider which allows us to connect to other types of databases like Access and Oracle.

ASP.NET (Building Web Applications)

Web Forms pages are built with ASP.NET technology. ASP.NET is a unified Web platform that provides all the services necessary for you to build enterprise-class applications.

ASP.NET is built on the .NET framework, so the entire framework is available to any ASP.NET application. Your applications can be authored in any language compatible with the common language runtime, including Microsoft Visual Basic, Visual C# and JScript.NET.

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework. The new framework will make use of the new open-source .NET Compiler Platform (code-name "Roslyn") and be cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6. The project is called ASP.NET vNext.

Life Cycle of ASP.NET

ASP.NET life cycle specifies, how:

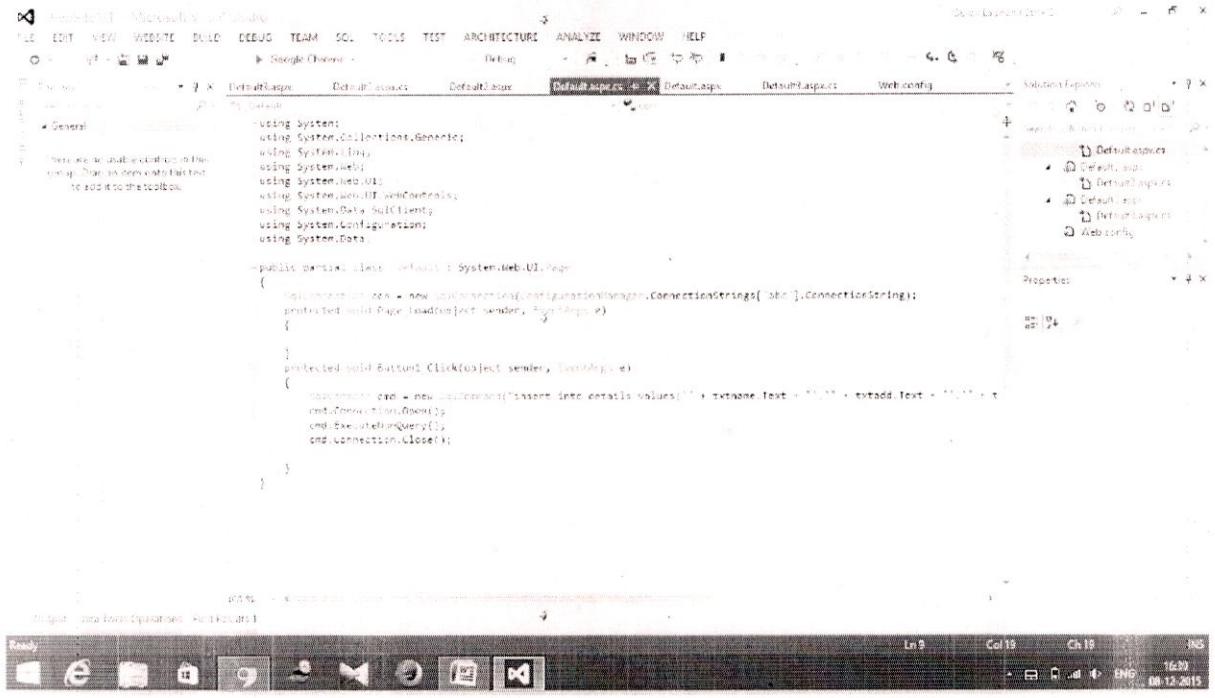
- ASP.NET processes pages to produce dynamic output

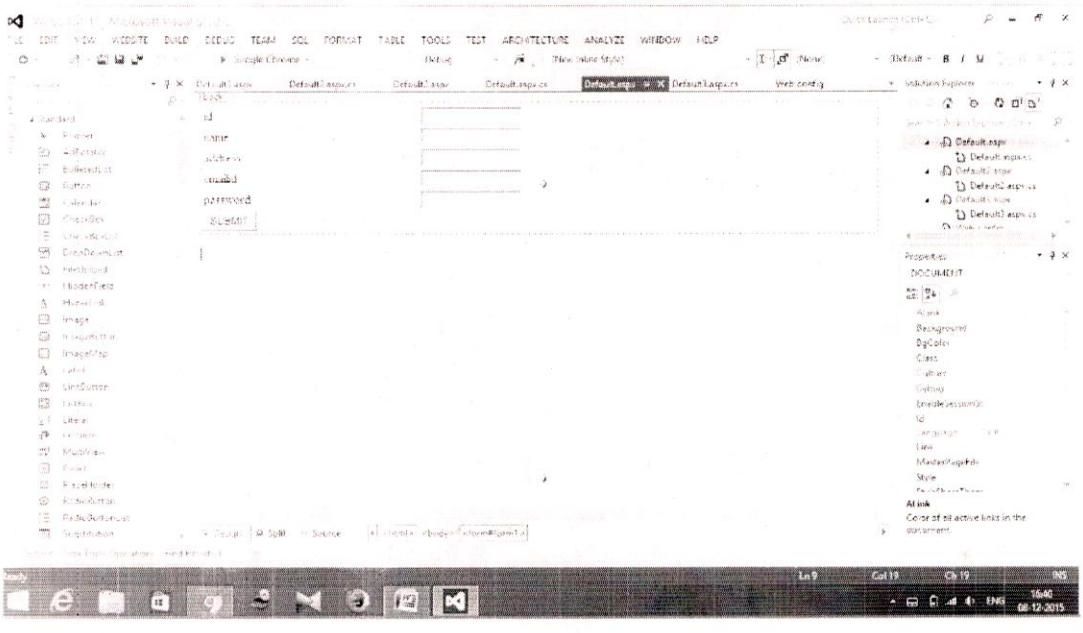
- The application and its pages are instantiated and processed
- ASP.NET compiles the pages dynamically

The ASP.NET life cycle could be divided into two groups:

- Application Life Cycle
- Page Life Cycle

An Example: Registration Page using .NET





Basic Controls

Button Controls

ASP.NET provides three types of button control:

- **Button** : It displays text within a rectangular area.
- **Link Button** : It displays text that looks like a hyperlink.
- **Image Button** : It displays an image.

When a user clicks a button, two events are raised: Click and Command.

Basic syntax of button control:

```
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Click" / >
```

Text Boxes and Labels

Text box controls are typically used to accept input from the user. A text box control can accept one or more lines of text depending upon the settings of the TextMode attribute.

Label controls provide an easy way to display text which can be changed from one execution of a page to the next. If you want to display text that does not change, you use the literal text.

Basic syntax of text control:

```
<asp:TextBox ID="txtstate" runat="server" ></asp:TextBox>
```

Check Boxes and Radio Buttons

A check box displays a single option that the user can either check or uncheck and radio buttons present a group of options from which the user can select just one option.

To create a group of radio buttons, you specify the same name for the GroupName attribute of each radio button in the group. If more

than one group is required in a single form, then specify a different group name for each group.

If you want check box or radio button to be selected when the form is initially displayed, set its Checked attribute to true. If the Checked attribute is set to true for multiple radio buttons in a group, then only the last one is considered as true.

Basic syntax of check box:

```
<asp:CheckBox ID= "chkoption" runat= "Server">
</asp:CheckBox>
```

Basic syntax of radio button:

```
<asp:RadioButton ID= "rdboption" runat= "Server">
</asp: RadioButton>
```

Validators

ASP.NET validation controls validate the user input data to ensure that useless, unauthenticated, or contradictory data don't get stored.

ASP.NET provides the following validation controls:

- RequiredFieldValidator
- RangeValidator
- CompareValidator
- RegularExpressionValidator
- CustomValidator
- ValidationSummary

AdRotator

The AdRotator control randomly selects banner graphics from a list, which is specified in an external XML schedule file. This external XML schedule file is called the advertisement file.

The AdRotator control allows you to specify the advertisement file and the type of window that the link should follow in the AdvertisementFile and the Target property respectively.

The basic syntax of adding an AdRotator is as follows:

```
<asp:AdRotator  runat = "server" AdvertisementFile = "adfile.xml"
Target =  "_blank" />
```

Multi Views

MultiView and View controls allow you to divide the content of a page into different groups, displaying only one group at a time. Each View control manages one group of content and all the View controls are held together in a MultiView control.

The MultiView control is responsible for displaying one View control at a time. The View displayed is called the active view.

The syntax of MultiView control is:

```
<asp:MultView ID= "MultiView1" runat= "server">
</asp:MultiView>
```

The syntax of View control is:

```
<asp:View ID= "View1" runat= "server">
</asp:View>
```

However, the View control cannot exist on its own. It would render error if you try to use it stand-alone. It is always used with a Multiview control as:

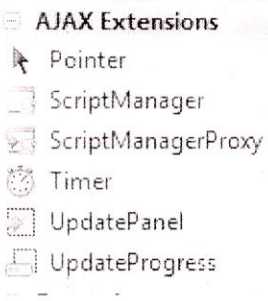
```
<asp:MultView ID= "MultiView1" runat= "server">
  <asp:View ID= "View1" runat= "server"> </asp:View>
</asp:MultiView>
```

Ajax Control

AJAX stands for Asynchronous JavaScript and XML. This is a cross platform technology which speeds up response time. The AJAX server controls add script to the page which is executed and processed by the browser.

However like other ASP.NET server controls, these AJAX server controls also can have methods and event handlers associated with them, which are processed on the server side.

The control toolbox in the Visual Studio IDE contains a group of controls called the 'AJAX Extensions'

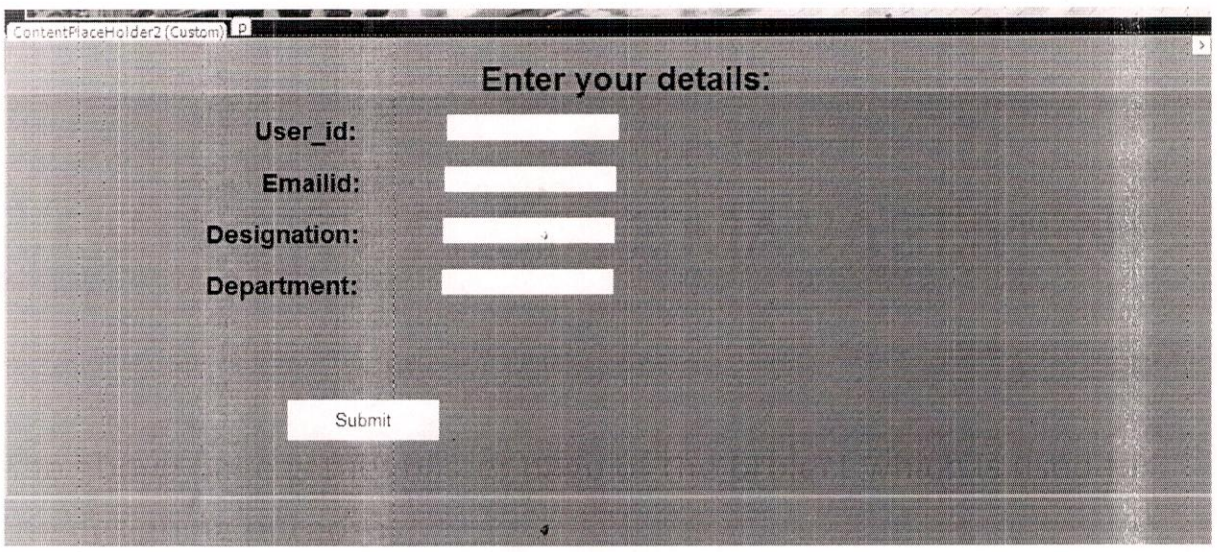


EMPLOYEE PROFILE MANAGEMENT

```
Client Objects & Events                                    (No Events)
  <meta name="description" content="" />
  <link href="styles.css" rel="stylesheet" type="text/css" />
  <link rel="stylesheet" href="nivo-slider.css" type="text/css" media="screen" />
      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server"></asp:ContentPlaceHolder>
  </head>
<body>
<form id="form1" runat="server">
<div id="bg">
<div id="bg_top_bg">
<div id="bg_top">
<div id="bg_bot">


<div id="main">
<!-- header begins -->
<div id="header">
    <div id="buttons">
      <a href="submit.aspx" class="but but_t"  title="">Sign in</a>
      <a href="Login.aspx" class="but" title="">Log in</a>
      <a href="gallery.html"  class="but" title="">Gallery</a>
      <a href="about_us.html"  class="but" title="">About us</a>
      <a href="contact_us.html" class="but" title="">Contact us</a><br />
        <br />
        NATIONAL INSTITUTE OF HYDROLOGY, ROORKEE<br />
        <br />
        <br />
    </div>

</div>
<!-- header ends -->
<div class="top">
```

The above page is the master page of the project which is used for the designing purposes.

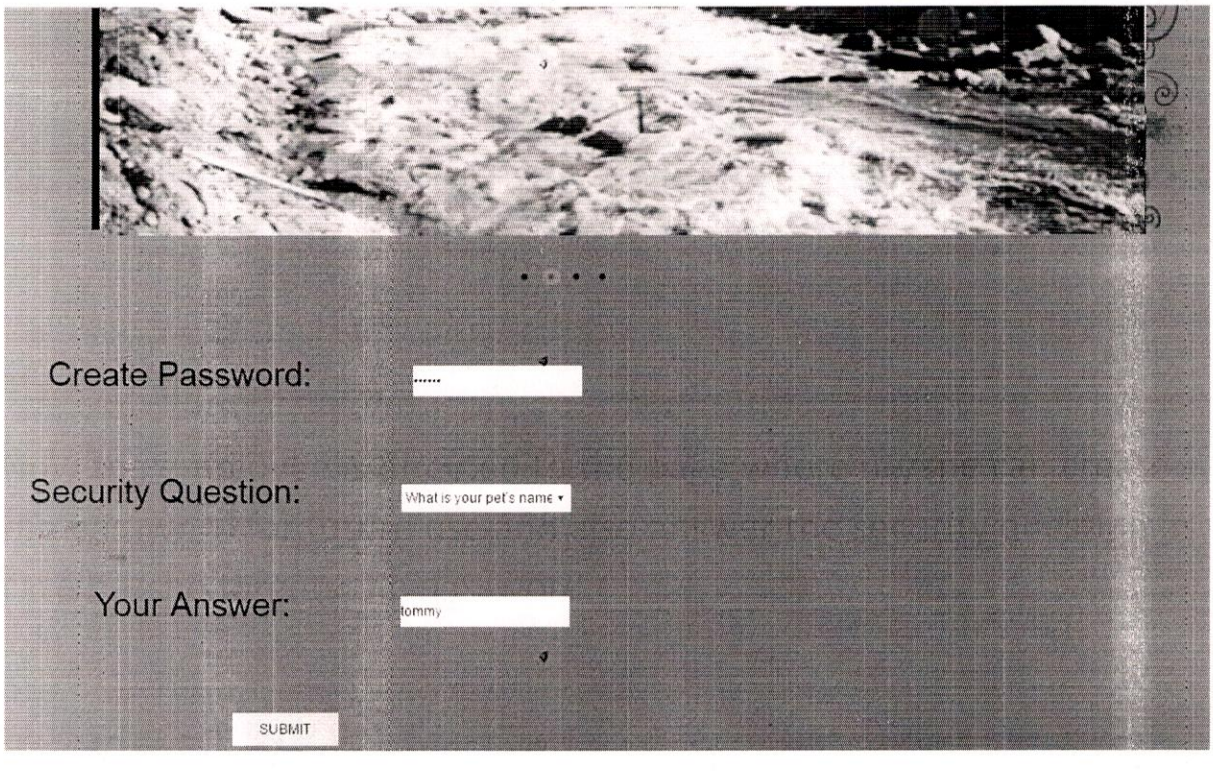
All the employees have a unique employee_id using which the employee can log in and enter the desired details.



The code for the above snippet of the output is as follows which is used to enter the unique user_id of the employee and is used for the sign up purpose.

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

public partial class login : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["abc"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlDataAdapter da = new SqlDataAdapter("select * from reg where user_id='" + TextBox1.Text + "' and emailid='" +
        DataTable dt = new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            Session["uid"] = dt.Rows[0].ItemArray[0].ToString();
            Response.Redirect("create.aspx");
        }
        else
        {
            Label1.Text = "invalid credentials";
        }
```

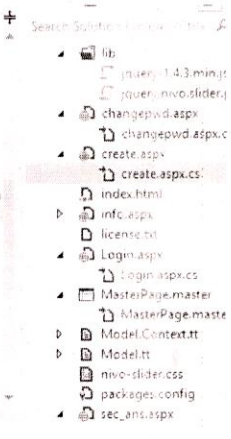
Now the employee is directed to the page where he/she can create the password for future use and also answer the security question in case he/she forgets the password.



The coding part for the above is:

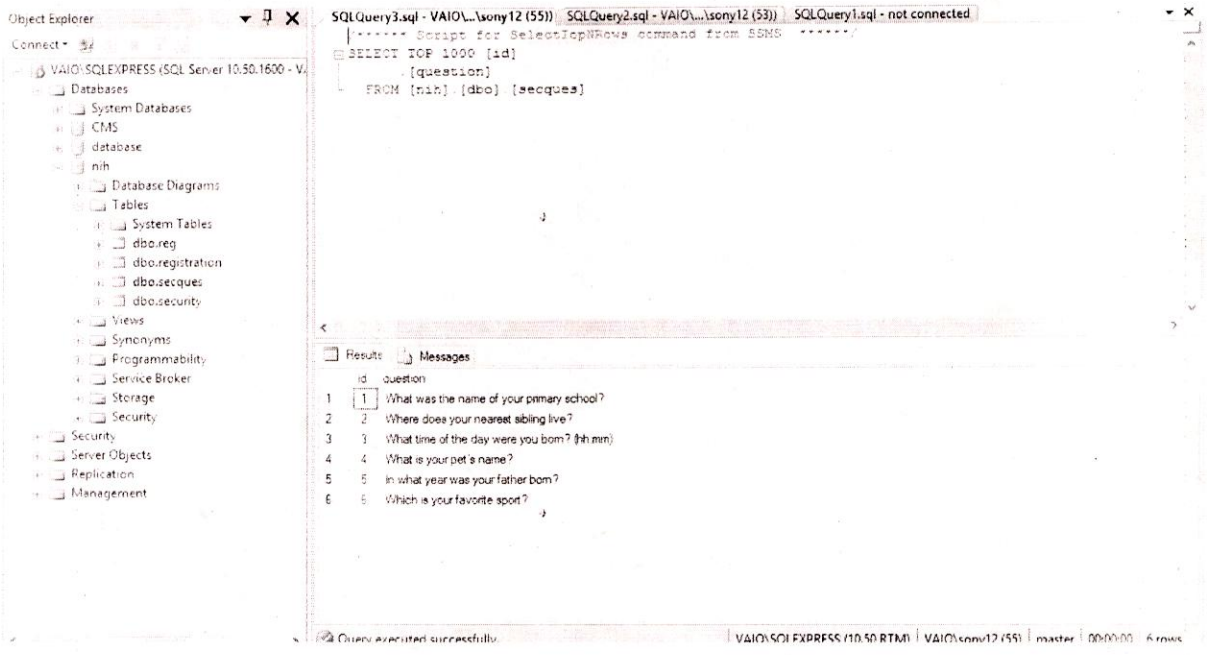
```
using System.Configuration;
using System.Data.SqlClient;
using System.Data;
//using System.Web.Security;

public partial class create : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["abc"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            SqlDataAdapter da = new SqlDataAdapter("select * from secques", con);
            DataTable dt = new DataTable();
            da.Fill(dt);
            DropDownList1.DataValueField = "id";
            DropDownList1.DataTextField = "question";
            DropDownList1.DataSource = dt;
            DropDownList1.DataBind();
        }

    }
```

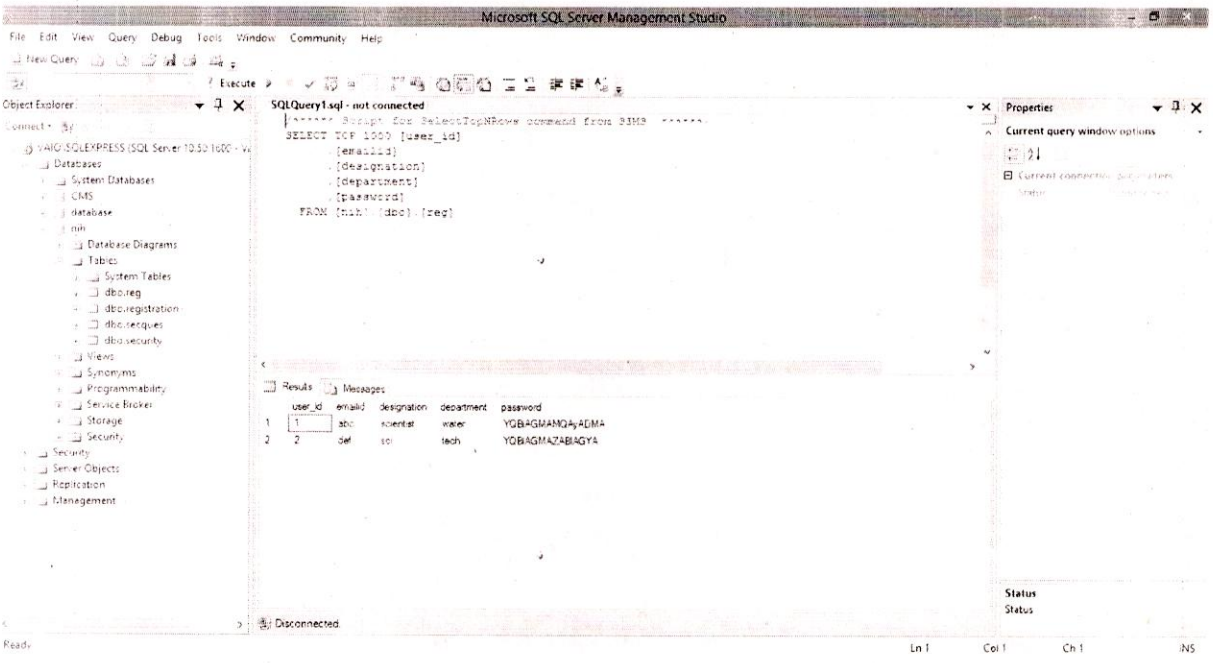
```
    }
    private string EncryptPassword(string pass)
    {
        byte[] bytes=System.Text.Encoding.Unicode.GetBytes(pass);
        string EncryptedPassword=Convert.ToBase64String(bytes);
        return EncryptedPassword;
    }
    protected void Button1_Click(object sender, EventArgs e)
    {

        SqlCommand cmd = new SqlCommand("update reg set password='"+EncryptPassword(TextBox1.Text)+"' where user_id='"+Session
        SqlCommand cmnd = new SqlCommand("update security set sec_ques='"+DropDownList1.SelectedItem.Text+"',sec_ans='" + Text
        cmd.Connection.Open();
        cmd.ExecuteNonQuery();
        cmd.Connection.Close();
        cmnd.Connection.Open();
        cmnd.ExecuteNonQuery();
        cmnd.Connection.Close();
        Response.Redirect("Login.aspx");
    }
}
```

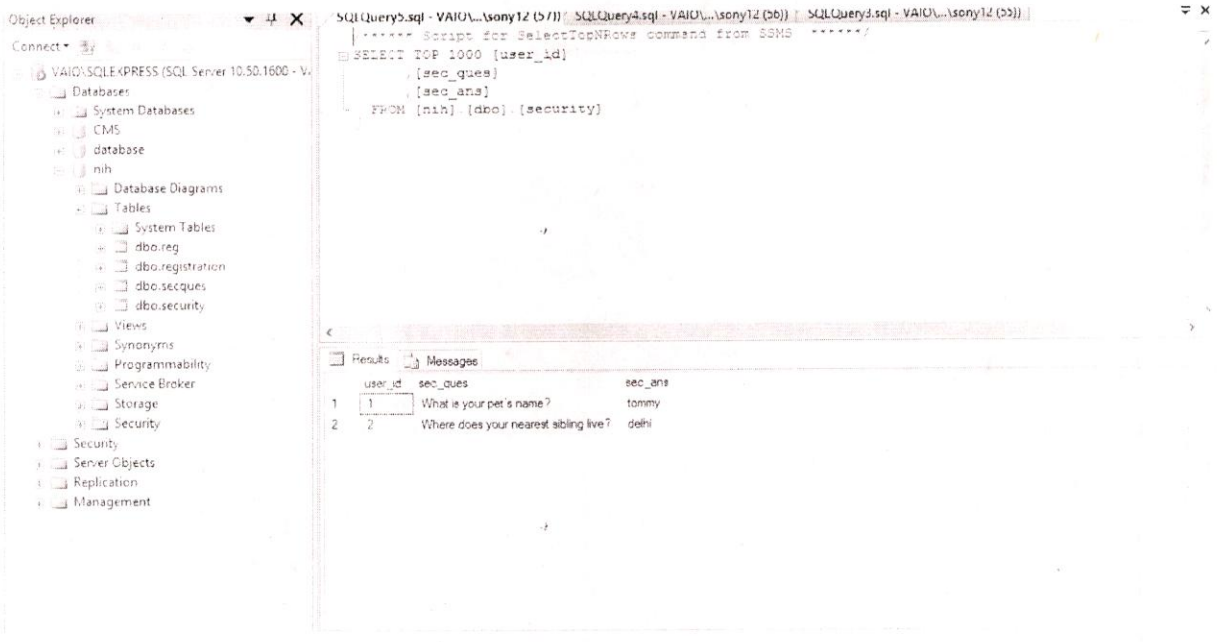


The password that the employee is entering is inserted into the database in an encrypted form which can be seen in the below screenshot of the database.

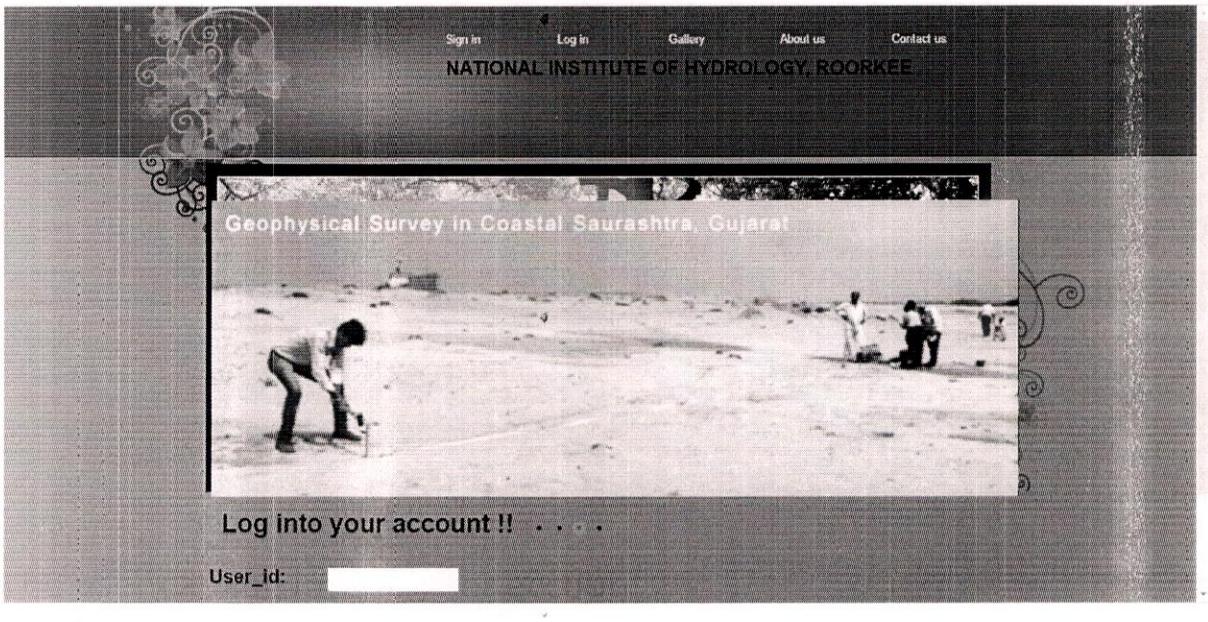
Also, Sessions are used enormously to take the information from one page to the other.

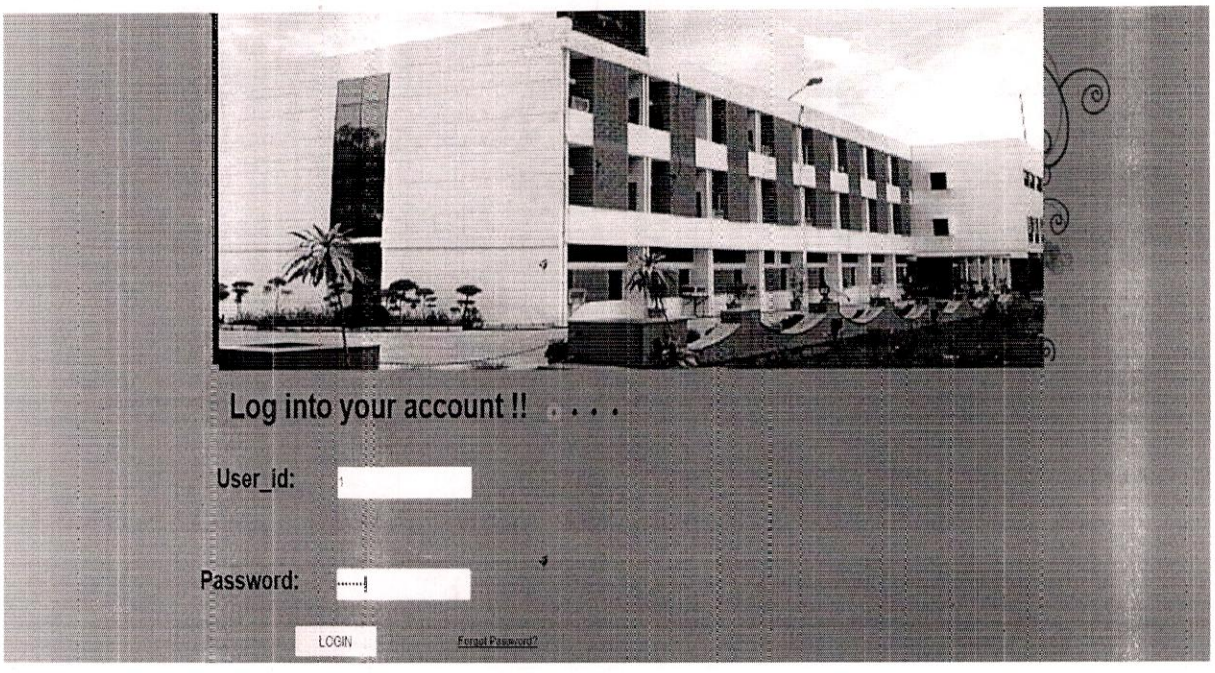

Object Explorer
Connect
VAIO\SQLEXPRESS (SQL Server 10.50.1600 - V
Databases
System Databases
CMS
database
nih
Database Diagrams
Tables
System Tables
dbo.reg
dbo.registration
dbo.secques
dbo.security
Views
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
Management
SQLQuery3.sql - VAIO\...\sony12 (55))
SQLQuery2.sql - VAIO\...\sony12 (53))
SQLQuery1.sql - not connected
SELECT TOP 1000 [id]
[question]
FROM [nih].[dbo].[secques]
Results
Messages
id question
1 1 What was the name of your primary school?
2 2 Where does your nearest sibling live?
3 3 What time of the day were you born? (hh:mm)
4 4 What is your pet's name?
5 5 In what year was your father born?
6 6 Which is your favorite sport?


Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query
Execute
Object Explorer
SQLQuery1.sql - not connected
SELECT TOP 1000 [user_id]
[emailid]
[designation]
[department]
[password]
FROM [nih].[dbo].[reg]
Properties
Current query window options
Results
Messages
user_id emailid designation department password
1 1 abc scientist water YQBAGMANQAyADMA
2 2 def scr tech YQBAGMAZABAGYA
Status
Disconnected
Ready
Ln 1 Col 1 Ch 1 INS



The user can log into his/her account.

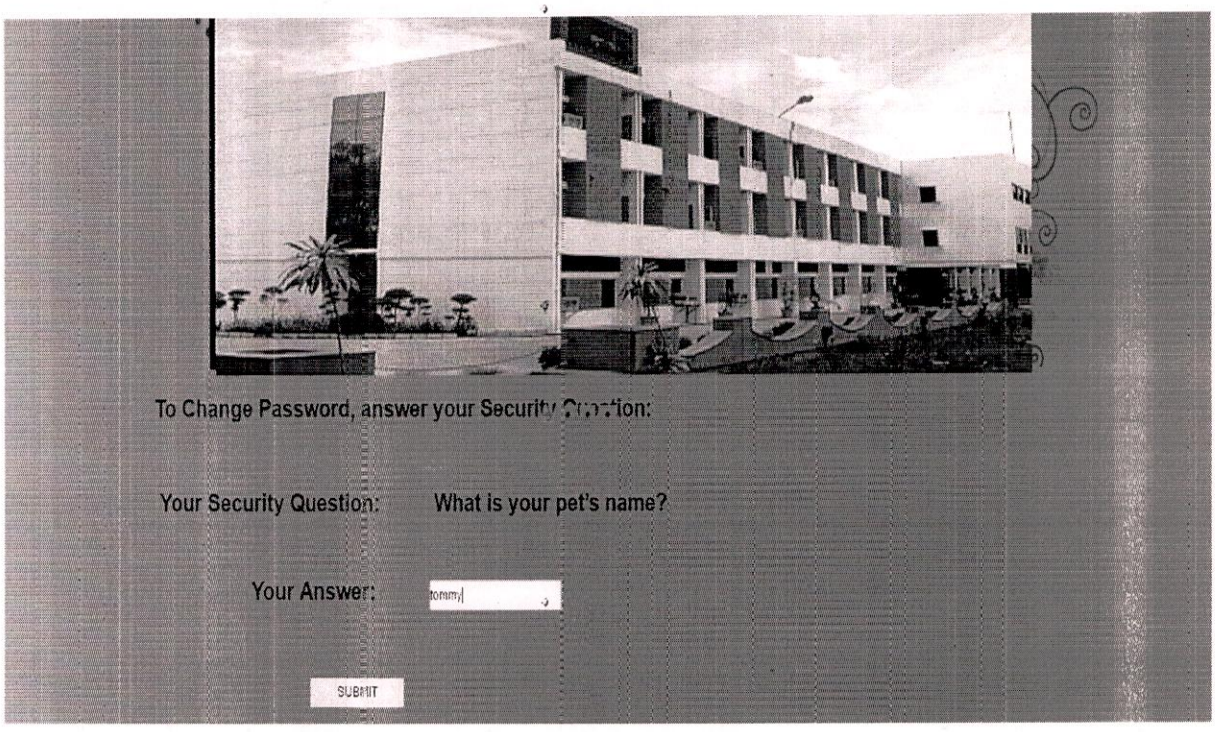


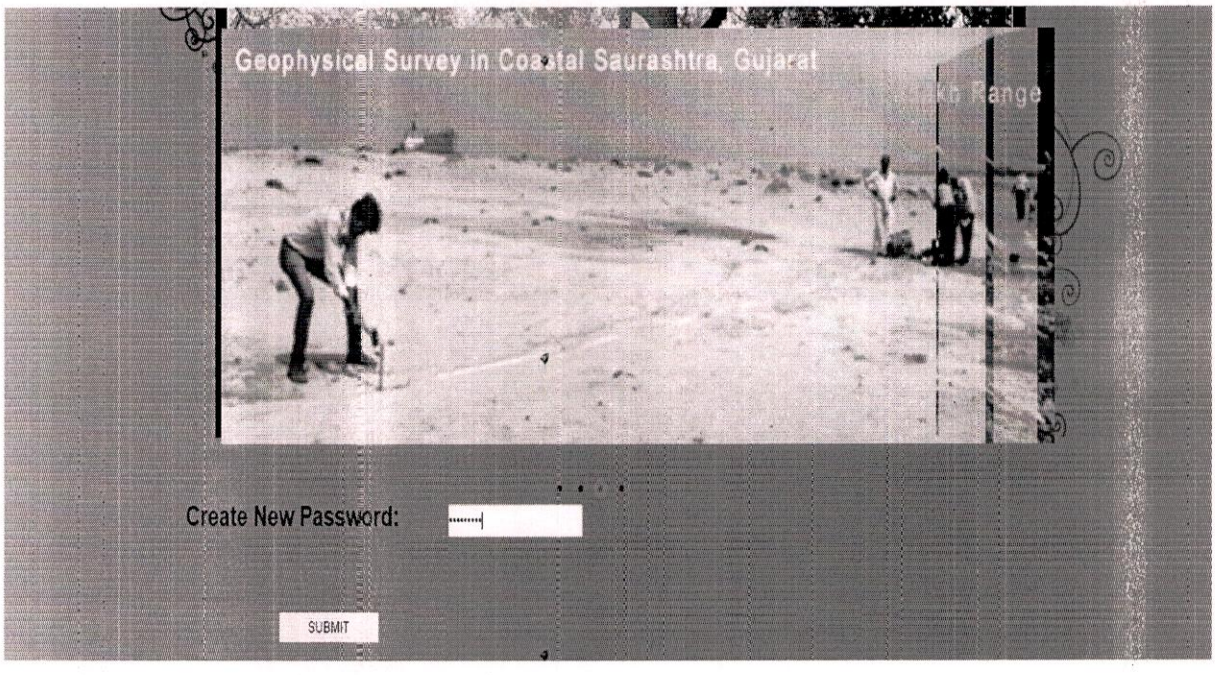



```
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;

public partial class Login : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["abc"].ConnectionString);
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    /* private String DecryptPassword(string Encryptedpass)
    {
        byte[] bytes = Convert.FromBase64String(Encryptedpass);
        string DecryptedPassword = System.Text.Encoding.Unicode.GetString(bytes);
        return DecryptedPassword;
    }*/
    private String EncryptPassword(string pass)
    {
        byte[] bytes = System.Text.Encoding.Unicode.GetBytes(pass);
```

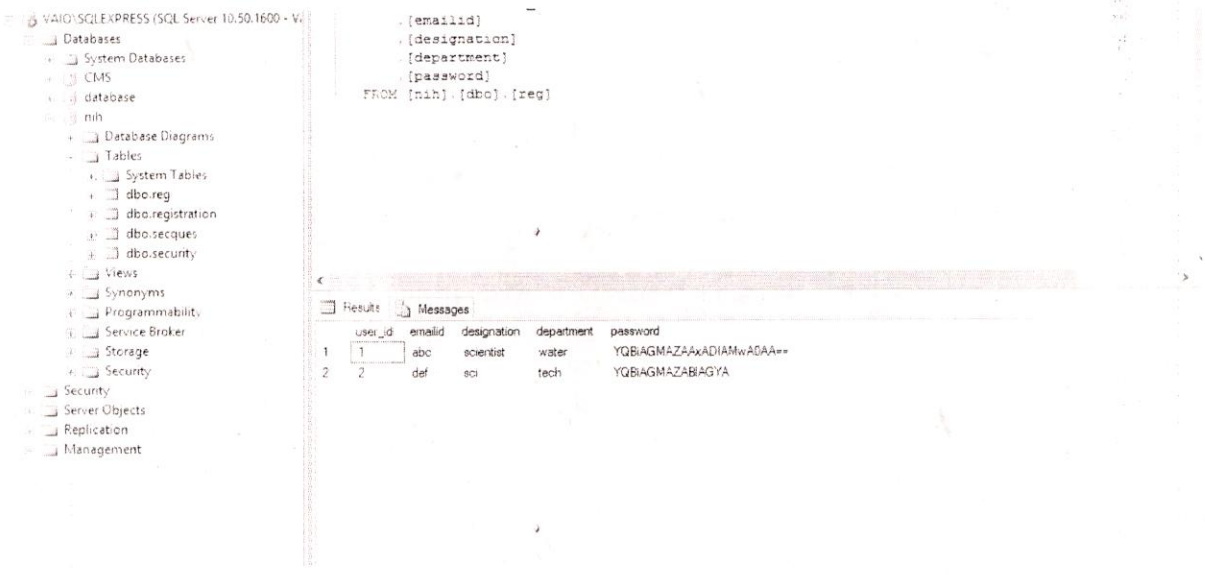
```
}
private String EncryptPassword(string pass)
{
    byte[] bytes = System.Text.Encoding.Unicode.GetBytes(pass);
    string EncryptedPassword = Convert.ToBase64String(bytes);
    return EncryptedPassword;
}
protected void Button1_Click(object sender, EventArgs e)
{   //string pwd=string.Empty;
    //SqlCommand cmd = new SqlCommand("select password from reg");
    //SqlDataReader dr = cmd.ExecuteReader();
    //pwd = dr[0].ToString();
    SqlDataAdapter da = new SqlDataAdapter("select * from reg where user_id='" + TextBox1.Text + "' and password='"+Encryp
    DataTable dt = new DataTable();
    da.Fill(dt);
    if (dt.Rows.Count > 0)
    {
        Response.Redirect("info.aspx");
    }
    else
    {
        Label1.Text = "invalid credentials";
    }
```

Many times the user can forget his/her password. Therefore, to change the password user just needs to answer the security question that he/she had chosen during the sign up and thus by answering the question correctly the user can change his/her password and can successfully log into the account and enter the details about his qualification,etc.

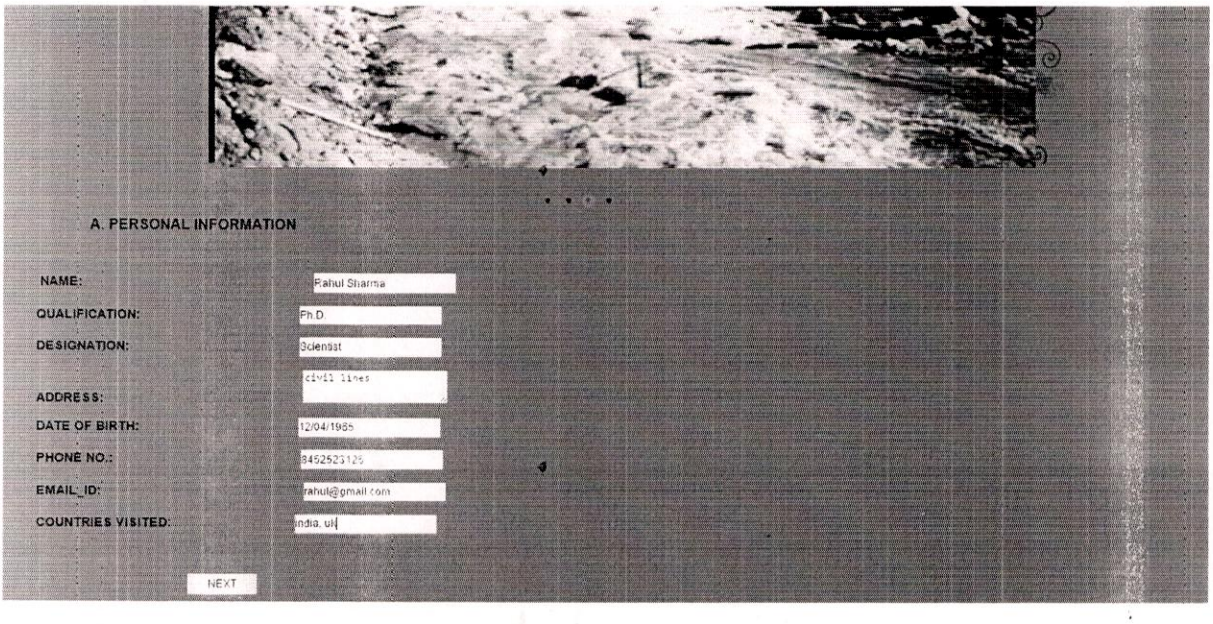


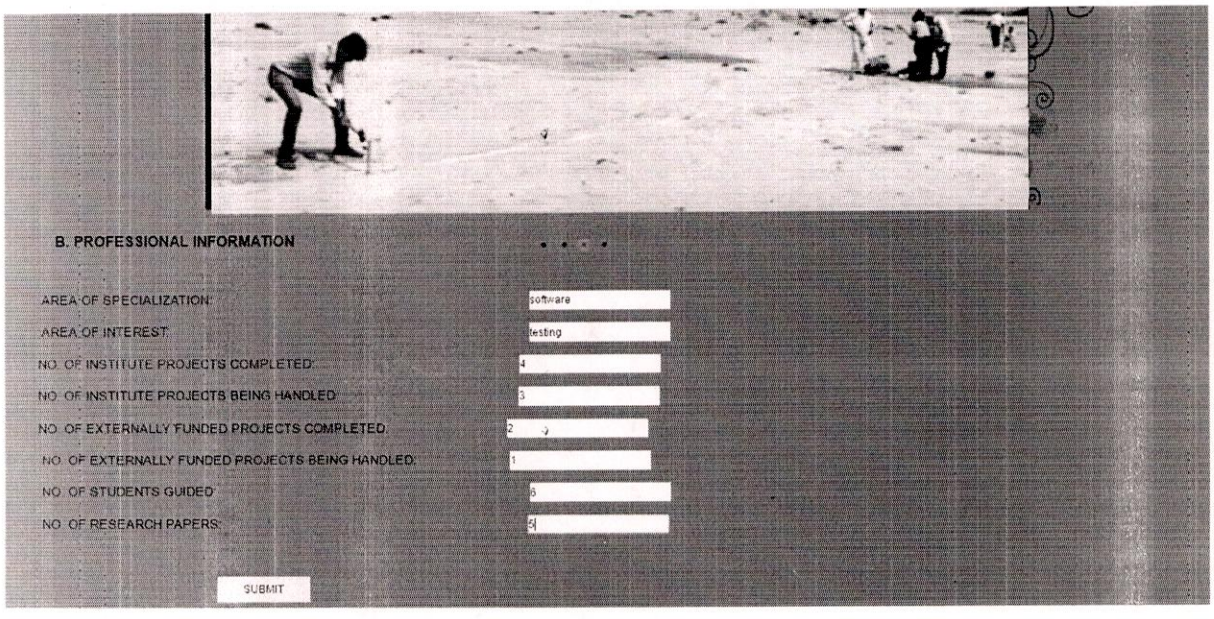


Hence, the password is changed and the new password is also represented in the encrypted form in the database.

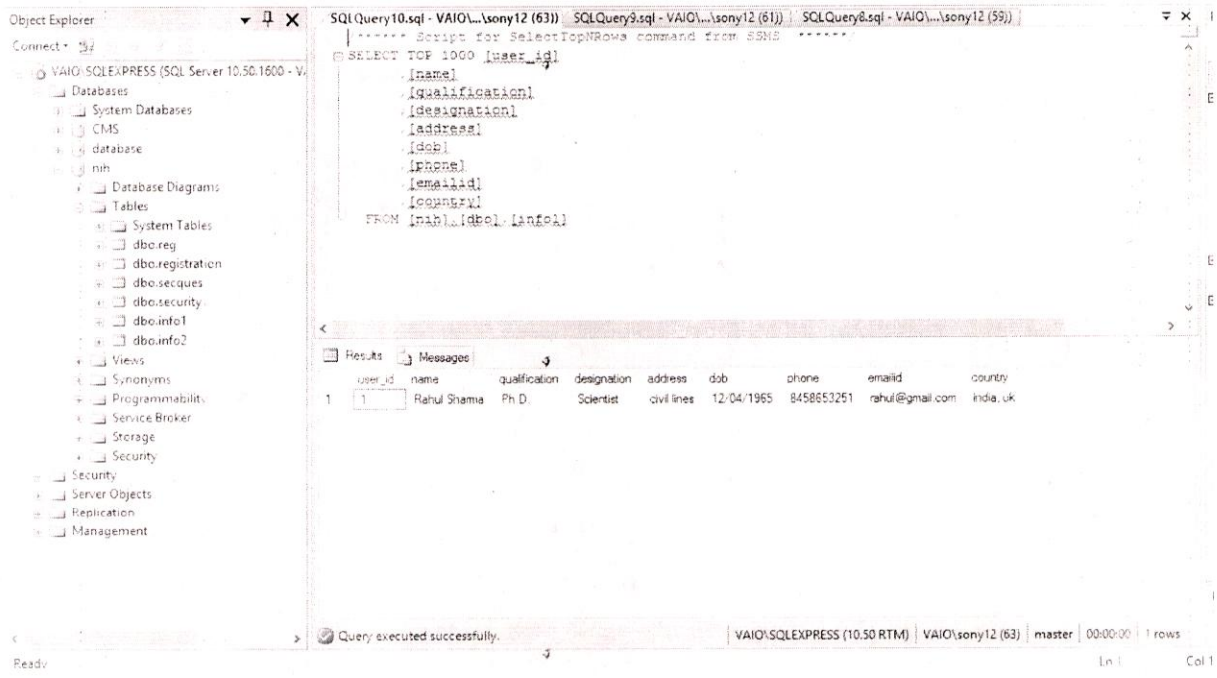


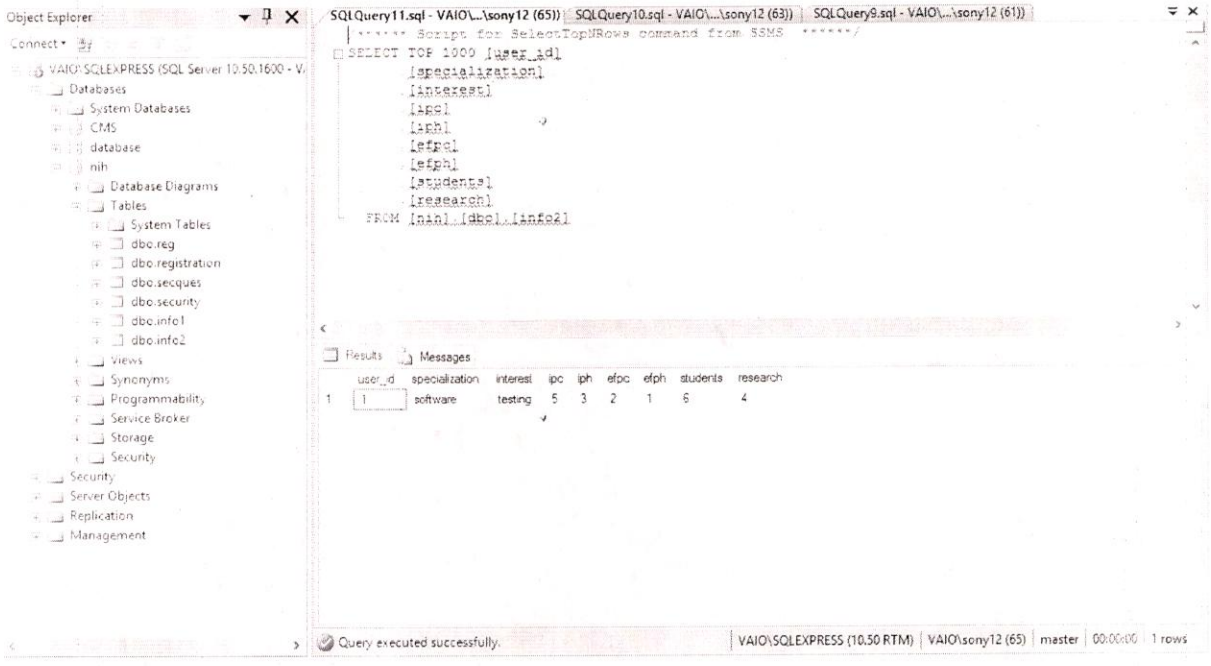
The Employee after logging into his/her account can manage his details accordingly can directly change without any need of the administrator. He/She can change both the personal details and the professional details.



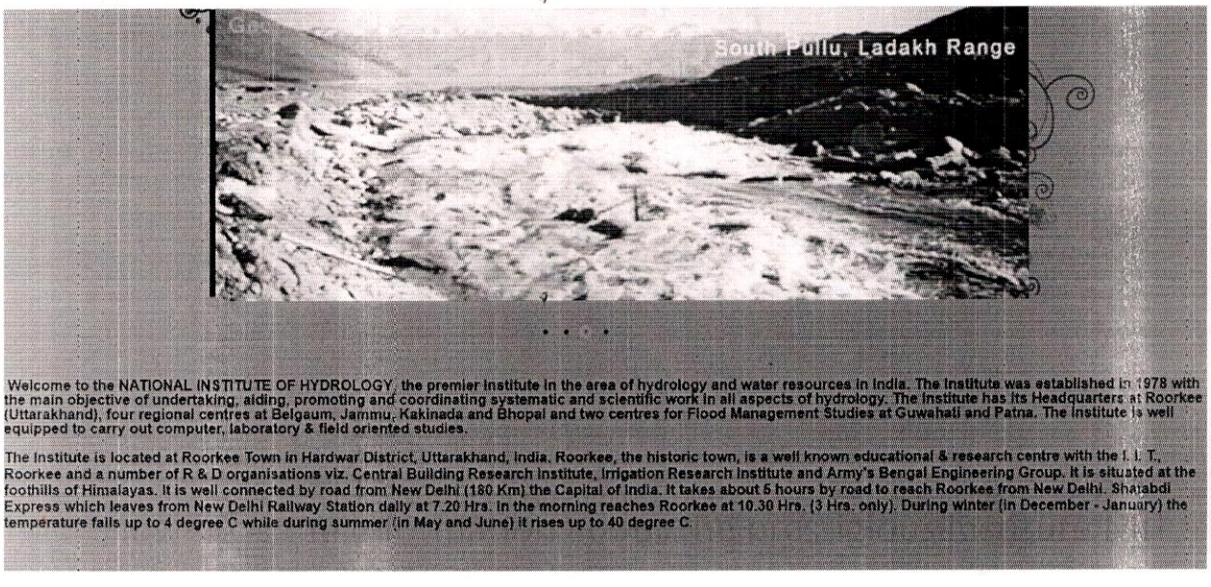


The corresponding changes made by the employee will be reflected in the database.





The following is the information about the prestigious institute NIH upon clicking the About Us link.



On clicking the Contact Us page, the following information comes in use.

Senior Administrative Officer
National Institute of Hydrology
Roorkee - 247667
Uttarakhand, India
Phone : +91-1332-272108, Fax : +91-1332-272123
Email : sao[at]nih.ernet.in

Website related information or any other information:
Web Information Manager
National Institute of Hydrology
Roorkee - 247667
Uttarakhand, India
Email : wim[at]nih.ernet.in

REFERENCES

- www.w3schools.com
- www.youtube.com
- www.wikipedia.com
- www.stackoverflow.com
- https://msdn.microsoft.com
- asp.net-tutorials.com