



# Mac Id Authentication for HR Employee Information

[ Master Project - 2007 ]

*Under the guidance of*

**Prof. Gonhsin Liu**

*Submitted by:*

**MANOJ DUVVA  
(ID#0734134)**



## Abstract

Today the web technology is growing rapidly and so with it the threat of intruder hacking the system. Companies to day depend hugely on web technology for sharing important information inside the companies. One such information system is employee information .for example Information like his personnel contacts, his qualification, his performance details etc. so net becomes a very unsafe place to host such information on system.

So it's our responsibility to think one step ahead. All computers has a Unique Mac-address .So if we can trace what is Mac address of the client system then we can make sure it system trying to access the site is valid or not. For example suppose if an unauthorized person gets valid login and password of such site even then he can't access the system unless he login from authorized machine.

This is nothing but to make sure that both the machine and the login user both are authorized.

So I am planning to develop a web application which has all contains all Human Recourse related information of all the employees in an organization and implement Mac-address base authentication in this site.

### **HARDWARE REQUIREMENTS: -**

- Pentium Min. 233 MHz.
- 32 MB Ram
- 512 KB Cache Memory
- Hard disk 4.3 GB
- Microsoft Compatible 101 or more Key Board

### **SOFTWARE REQUIREMENTS: -**

Operating System	:	Windows 98/NT/XP
Back End	:	MS SQL SERVER &MS EXCELSHEET
Front End (ASP.NET)	:	Microsoft Visual studio 2005
Database Connectivity	:	Microsoft ODBC Driver.



## Overview:

This project deals with the collection of HR employee information (database). But this is a secured site so that if any one even knows the login id and the password but he cannot open this any other machine. Here our system's Mac Id is authenticated with this site. The same login id and the password in the same machine can only be operated.

Here we can create a role such a ways that his user id and password and his Mac id is required to create a role. In this way the user is created.

Even this site has different roles for a user.

A user can be given different permission such that he can only read the data but not edit the data. Even a user can be given permission on the access of only specific pages.

Here the data is imported for an Excel page. First the date is entered in an formatted Excel sheet. Then through this project it can be edited and deleted depending the user logged on and by submitting the changes it effects in the database.

### **Learnt from this Project:**

Visual studio.net 2005

Sql server 2000

Database

Analyzing a Problem



## **Introduction:**

### **What is a MAC Address**

A MAC address is a unique number assigned to a Network Interface Card (NIC), commonly called an Ethernet card. This "address" is created by the manufacturer. A MAC address is a 12-digit number. Each digit is a number from 0-9 or a letter from A-F. Sometimes the digits of a MAC address are separated by colons or dashes. Examples of possible MAC addresses include: 080007A92BFC, 09:00:07:A9:B2:EB, or 09-10-4A-B9-E2-A4.

### **In Windows, how do I check my computer's IP address or physical (MAC) address?**

#### **Windows NT, 2000, and XP**

To see what IP address, default gateway, and subnet mask your computer is using:

1. Click the **Start** button and select **All Programs** (XP) or **Programs** (NT and 2000).
2. Select **Accessories**, then **Command Prompt** (2000 and XP), or **Command Prompt** (NT).
3. At the command prompt, enter: ipconfig
4. To close the command prompt, enter: exit



For other IP information on Windows, including the physical address, at the command prompt, enter:

### **ipconfig /all | more**

This will give you information regarding your computer's TCP/IP setup, including:

- Host name
- DNS servers
- Physical address
- IP address
- Subnet mask
- Default gateway
- WINS servers

The IP address will be a series of numbers following the pattern xxx.xxx.xxx.xxx, for example, 156.56.27.32. The physical address, which many people call the MAC address, will be a series of six hexadecimal number pairs following the pattern xx-xx-xx-xx-xx-xx, for example, 00-40-A4-F3-C0-01.

### **Windows XP, without using the command prompt**

Alternatively, in Windows XP, if you want to find this information and want to stay within the graphic user interface, or for any reason do not want to use DOS commands, you can find this information by doing the following:

1. Right-click the **My Network Places** icon, and then select **Properties**.
2. Double-click the **Local Area Connection** icon. Occasionally, there will be a number on the end (e.g., **Local Area Connection 2**); it depends on your computer's configuration. If you are looking for the MAC address for



a wireless adapter, the icon will normally be named **Wireless Network Connection**.

3. Click the **Support** tab.
4. Click the **Details...** button.

The physical address (MAC address) and IP address will be listed in the window that appears.

### **Windows 95, 98, and Me**

To check your computer's IP address or physical address, follow the steps below:

1. Click the **Start** button and select **Run...**
2. Type winipcfg and click **OK**.
3. From the drop-down menu, select the adapter being used for your connection:
  - Select **PPP Adapter** if you're dialing in.
  - Select the Ethernet adapter if you're connected to a LAN or a broadband connection.
  - Select the wireless adapter if you're connected via wireless.

The IP address will appear in its labeled field. The MAC address will appear in a field labeled **Adapter Address**.

Alternatively, you may follow these steps:

1. Click the **Start** button and select **Settings**, then **Control Panel**.
2. Double-click **Network**. Select **TCP/IP**, then click **Properties**.

The IP address appears under the **IP Address** page tab.

---



## About Database Access:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. If your database isn't stored on a computer, or only parts of it are, you may be tracking information from a variety of sources that you're having to coordinate and organize yourself.

Using Microsoft Access, you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables; view, add, and update table data using online forms; find and retrieve just the data you want using queries; and analyze or print data in a specific layout using reports.

To store your data, create one table for each type of information you track. To bring the data from multiple tables together in a query, form, or report, you define relationships between the tables.

To find and retrieve just the data that meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.

## **Import Data OR Link Data :**

Microsoft Access can import or link table data from other Microsoft Access databases, as well as data from other programs and file formats, such as Microsoft Excel, dBASE, Microsoft FoxPro, or Paradox. You can also import or link



(read-only) HTML tables and lists, which can reside on your local computer, a network server, or an Internet server.

Importing data creates a copy of its information in a new table in your Microsoft Access database. The source table or file is not altered in this process. Linking data enables you to read and in most cases update data in the external data source without importing. The external data source's format is not altered so that you can continue to use the file with the program that originally created it, but you can add, delete, or edit its data using Microsoft Access as well.

When importing data, you can't append data to existing tables (except when importing spreadsheet or text files). However, once you have imported a table, you can perform an append query to add its data to another table.

### ***Exporting Data:***

Export data from a table or query to many different formats. You can also export any of your database objects to another Microsoft Access database. This is effectively the same as copying and pasting objects between databases.

You can export tables, queries, forms, and reports to HTML files, or tables, queries, and forms to IDC/HTX files, which you can then use as part of a World Wide Web site.

### **TABLES:**

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means we can store that data only once, which makes your database more efficient and reduces data-entry errors. Tables organize data into columns and rows.





In table Datasheet view, you can add, edit, or view the data in a table. You can also check the spelling and print your table's data, filter or sort records, change the datasheet's appearance, or change the table's structure by adding or deleting columns.

In table Design view, you can create an entire table from scratch, or add, delete, or customize an existing table's fields.

### **QUERIES:**

Use queries to view, change, and analyze data in different ways. You can also use them as the source of records for forms and reports.

The most common type of query is a select query. A select query retrieves data from one or more tables using criteria you specify, and then displays it in the order you want.

We create a query with a wizard or form scratch in query Design view. In Design view, we have to specify the data you want to work with by adding the tables or queries that contain the data, and then by filling in the design grid.

### **REPORTS:**

A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.

Most of the information in a report comes from an underlying table, query, or SQL statement, which is the source of the report's data. Other information in the report is stored in the report's design.

We can create the link between a report and its record source by using graphical objects called controls. Controls can be text boxes that display names



and numbers labels that display titles, and decorative lines that graphically organize the data and make the report more attractive.

**Macros:**

A macro is a set of one or more actions that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.

A macro can be one macro composed of a sequence of actions, or it can be a macro group. You can also use a conditional expression to determine whether in some cases an action will be carried out when a macro runs.



## **INTRODUCTION TO SQL**

Structured Query Language (SQL), is the set of commands That all programs and users must use to access data within the Oracle database. Application programs and Oracle tools often allow users to access the database without directly using SQL, but these applications in turn must use SQL when executing the user's request. This chapter provides background information on SQL used by most relational database systems. Topics include:

- History of SQL
- SQL Standards
- How SQL Works

### **History of SQL**

The paper, "A Relational Model of Data for Large Shared Data Banks," by Dr. E. F. Codd, was published in June 1970 in the Association of Computer Machinery (ACM) journal, Communications of the ACM. Codd's model is now accepted as the definitive model for relational database management systems (RDBMS). The language, Structured English Query Language (SEQUEL) was developed by IBM Corporation, Inc. to use Codd's model. SEQUEL later became SQL. In 1979, Relational Software, Inc. (now Oracle Corporation) introduced the first commercially available implementation of SQL. Today, SQL is accepted as the standard RDBMS language.

### **SQL Standards**

Oracle SQL complies with industry accepted standards. Oracle Corporation ensures future compliance with evolving SQL standards by actively involving key personnel in SQL standards committees. Industry accepted committees are the American National Standards Institute (ANSI) and the International Standards Organization (ISO), which is affiliated with the International Electrotechnical



Commission (IEC). Both ANSI and the ISO/IEC have accepted SQL as the standard language for relational databases. When a new SQL standard is simultaneously published by these organizations, the names of the standards conform to conventions used by the organization, but the technical details are exactly the same.

### **How SQL Works**

This section describes many of the reasons for SQL's widespread acceptance by relational database vendors as well as end users. The strengths of SQL benefit all ranges of users including application programmers, database administrators, management, and end users.

Technically speaking, SQL is a data sub language. That is to say, the purpose of SQL is to interface to a relational database such as Oracle, and all SQL statements are instructions to the database. In this it differs from general purpose programming languages like C and Basic. Among the features of SQL are the following:

- It processes sets of data as groups rather than as individual units.
- It provides automatic navigation to the data.
- It uses statements that are complex and powerful individually, and that therefore stand alone. Flow-control statements were not part of SQL originally, but they are found in the recently accepted optional part of SQL, ISO/IEC 9075-5 : 1996. Flow-control statements are commonly known as "Persistent Stored Modules" or PSM, and Oracle's PL/SQL extension to SQL is close to PSM. Essentially, SQL lets you work with data at the logical level, only being concerned with the implementation details when you want to manipulate them.

**SQL provides commands for a variety of tasks including:**



- querying data
- inserting, updating, and deleting rows in a table
- creating, replacing, altering, and dropping objects
- controlling access to the database and its objects
- guaranteeing database consistency and integrity

SQL unifies all of the above tasks in one consistent language.

### **Oracle and Standard SQL**

This appendix discusses the following topics:

- Oracle's conformance to the SQL standards established by industry standards governing bodies
- Oracle's extensions to standard SQL
- Locating extensions to standard SQL with the FIPS Flogger

### **Introduction to Data Definition Language**

The data Definition Definition Language is used to create an object, alter the structure of an object and also to drop the table created.

A table is a unit of storage which holds data in the form of rows and columns.

The DDL can be classified in to following categories.

- Create table command
- Alter table command
- Drop table command

#### **1. Create:**

This command is used to create a table in the data base.

**Create table <table name>(column definition1,column definition2,....);**



## 2. Alter command:

Alter command cater to the need of the following situations.

A) When the user wants to add a new column to the existing table.

**Alter table <table name> add (column definition...);**

B) When the user wants to modify the existing column definition.

**Alter table <table name> modify (column definition...);**

## 3. Drop table command:

This command is used to drop the table from the data base.

**Drop table <table name>;**

## DATA MANIPULATION LANGUAGE COMMANDS:

Data Manipulation Commands are most frequently used SQL commands.

They are as follows,

- Insert
- Select
- Update
- Delete

## Insert Command:

This command is used to to add one or more rows to the table. While using this command the values are seperated by commas and the datatypes char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined in the table.

**Insert into <table name> values (a list of data values);**



### **SELECT COMMAND:**

To perform a query we use the select command. The query is a request for the information. It is the most common data base operation used.

**Select column name ..... from <table name>;**

**Or**

**Select \* from <table name>;**

It will give all the information in the table.

### **UPDATE COMMAND:**

Update command is used to alter the column values in the table. The update command consists of a 'set' clause and an optional 'where' clause.

**Update <table name> set field=value,... where condition;**

**e.g.**

**update emp set sal=10000 where name='Reddy';**

### **Delete Command:**

This command is used to delete several rows from the table. After inserting rows in a table we can also delete then if required. The delete command consists of a 'from clause' followed by an optional 'where clause'.

**Delete from <table name> where conditions;**



## **Transaction control commands:**

### **Commit:**

This command is used to end a transaction.

**Commit work;**

*Or*

**Commit;**

### **Rollback:**

A rollback command is used to undo the work done in the current transaction.

**Rollback;**

## **GRANT**

The GRANT command (System Privileges and Roles) is an extension to standard SQL.

The GRANT command (Object Privileges) supports the following other privileges on other objects in addition to the DELETE, INSERT, REFERENCES, SELECT, and UPDATE privileges on tables and views supported by Entry SQL-92:

- ALTER
- EXECUTE
- INDEX
- READ

This command also supports granting object privileges to roles.

## **Operators**

This section describes additional operators and additional functionality of standard operators.

### **Additional Operators**

Oracle supports these operators that are not part of Entry SQL-92:





- || character operator (character concatenation)
- !=, ^=, and ¬= comparison operators (inequality)
- MINUS set operator
- INTERSECT set operator
- (+) operator (outer join)

## INTEGRITY CONSTRAINTS

An integrity constraint is a mechanism used by Oracle to prevent invalid data entry into the table. The following are the various types of integrity constraints.

- Domain integrity constraint
  - a) Not Null constraint
  - b) Check constraint
- Unique constraint
- Referential integrity constraint

### 1. Not Null :

When a 'Not Null' constraint is enforced on a column or a set of columns in a table, it will not allow Null values.

e.g.

```
create table <table name>(column definition1 constraint  
<constraint name>Not Null, column definition2,...);
```

### 2. Check constraint:

These are rules governed by logical or boolean expression.

e.g.

```
create table emp(eid number(4) constraint con check(eid>100));
```



### **3. Unique constraint:**

This constraint is used to prevent the duplication of values within the rows of a specified column in the table.

e.g.

**create table emp(eid number(4) constraint con unique);**

### **Primary Key Constraint:**

This constraint avoids duplication of rows and does not allow Null values, when enforced in a column or set of columns.

**create table emp(eid number(4) constraint con primary key);**

### **4. Referential Integrity Constraint:**

To establish a 'parent-child' or a 'master-detail' relationship between two tables having a common column, we make use of referential integrity constraints.

Basic components related to this constraint are:

#### **Foreign Key:**

A column or combination of columns include in the definition of referential integrity which would refer to a referenced key.

#### **Referenced Key:**

It is a unique or primary key which is defined on a column belonging to the parent table.

#### **Child Table:**

This table depends upon the values present in the referenced key of the parent table, which is referred by a foreign key.

#### **Parent table:**



This table determines insertion or updation of data can be done in child table.  
This table would be referred by child's table foreign key.

e.g.

```
create table emp(eid number(4) constraint con references  
dept(deptid));
```

### **On Delete Cascade clause**

If all the rows under the referenced key column in a parent table are deleted, then all rows in the child table with dependent foreign key column will also be deleted automatically.

e.g.

```
create table emp(eid number(4) constraint con references dept(deptid)  
on delete cascade);
```



## Overview of the .NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI)



## Mac Id Authentication For HR Employee Information

---

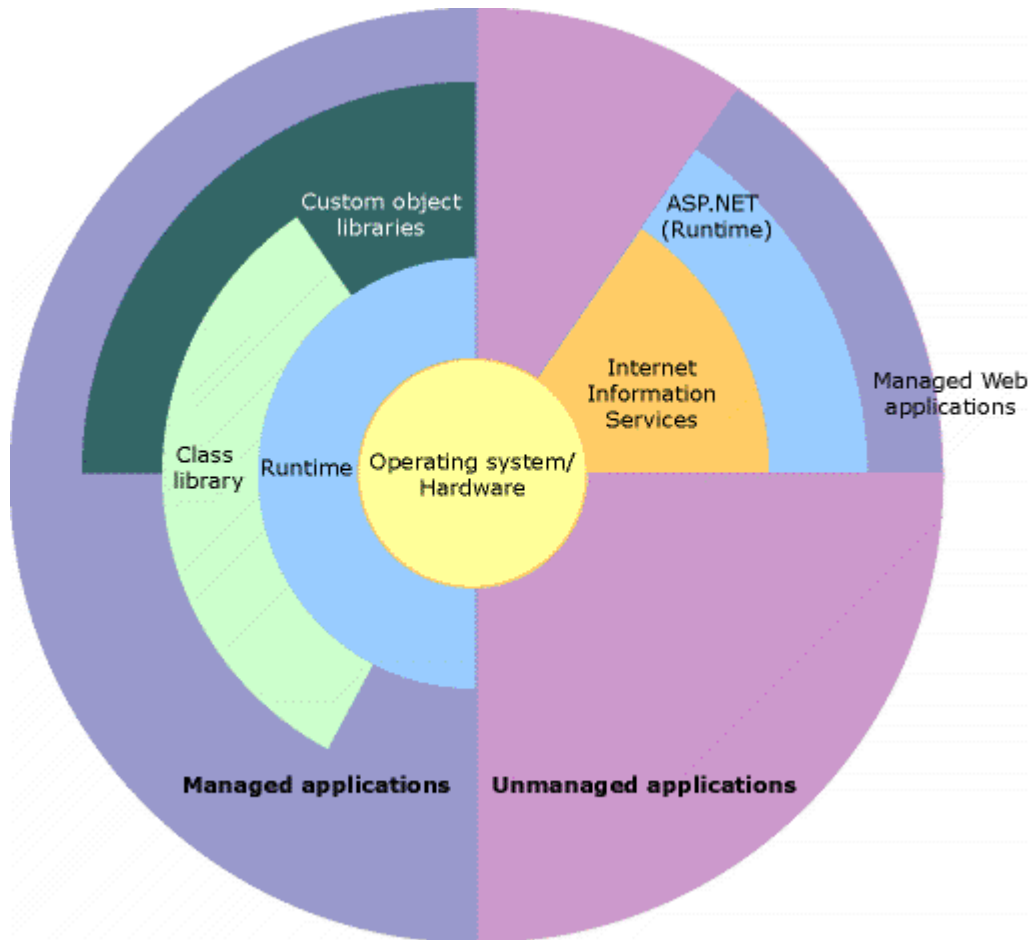
applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.



The following sections describe the main components and features of the .NET Framework in greater detail.

### **Features of the Common Language Runtime**

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive



## Mac Id Authentication For HR Employee Information

---

functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native



machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

### **.NET Framework Class Library**

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.





- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

### **Client Application Development**

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates

the developer interface, making coding simpler and more consistent.

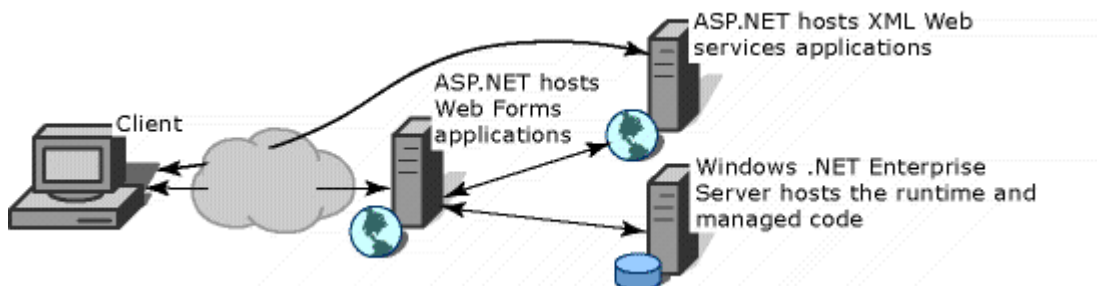
Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system (such as GUI elements and limited file access) without being able to access or compromise other resources. Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Your applications can implement the features of a local application while being deployed like a Web page.

## Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

### Server-side managed code



ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using



## Mac Id Authentication For HR Employee Information

---

managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived



from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

## Namespaces

Namespaces organize the objects defined in an assembly. Assemblies can contain multiple namespaces, which can in turn contain other namespaces. Namespaces prevent ambiguity and simplify references when using large groups of objects such as class libraries.

For example, Visual Studio .NET defines the **ListBox** class in the **System.Windows.Forms** namespace. The following code fragment shows how to declare a variable using the fully qualified name for this class:

```
Dim LBox As System.Windows.Forms.ListBox
```

Visual Studio .NET namespaces address a problem sometimes known as *namespace pollution*, in which the developer of a class library is hampered by the use of similar names in another library. These conflicts with existing components are sometimes called *name collisions*.

For example, if you create a new class named `Listbox`, you can use it inside your project without qualification. However, if you want to use the Visual Studio .NET **Listbox** class

in the same project, you must use a fully qualified reference to make the reference unique. If the reference is not unique, Visual Basic .NET produces an error stating that the name is ambiguous. The following code snippet demonstrates how to declare these objects:

```
' Define a new object based on your ListBox class.  
Dim LBC as New ListBox  
  
' Define a new Windows.Forms ListBox control.  
Dim MyLB as New System.Windows.Forms.ListBox
```

The following illustration shows two namespace hierarchies, both containing an object named ListBox.



By default, every executable file you create with Visual Basic .NET contains a namespace with the same name as your project. For example, if you define an object within a project named ListBoxProject, the executable file, ListBoxProject.exe, contains a namespace called ListBoxProject.

Multiple assemblies can use the same namespace. Visual Basic .NET treats them as a single set of names. For example, you can define classes for a namespace called SomeNameSpace in an assembly named Assemb1, and define additional classes for the same namespace from an assembly named Assemb2.

## Fully Qualified Names

Fully qualified names are object references that are prefixed with the name of the namespace where the object is defined. You can use objects defined in other projects if you create a reference to the class (by choosing **Add Reference** from the **Project** menu) and then use the fully qualified name for the object in your code. The following code fragment shows how to

to use the fully qualified name for an object from another project's namespace:



```
Dim LBC As New ListBoxProject.Form1.ListBox()
```

Fully qualified names prevent naming conflicts because the compiler can always determine which object is being used. However, the names themselves can get long and cumbersome. To get around this, you can use the **Imports** statement to define an *alias* — an abbreviated name you can use in place of a fully qualified name. For example, the following code snippet creates aliases for two fully qualified names, and uses these aliases to define two objects:

```
Imports LBCControl = System.Windows.Forms.ListBox
```

```
Imports MyListBox = ListBoxProject.Form1.ListBox
```

```
Dim LBC As LBCControl
```

```
Dim MyLB As MyListBox
```

If you use the **Imports** statement without an alias, you can use all the names in that namespace without qualification provided they are unique to the project. If your project contains **Imports** statements for namespaces that contain items with the same name, you must fully qualify that name when you use it. Suppose, for example, your project contained the following two **Imports** statements:

```
Imports MyProj1 ' This namespace contains a class called  
Class1.
```

```
Imports MyProj2 ' This namespace also contains a class  
called Class1.
```

If you attempt to use `Class1` without fully qualifying it, Visual Basic .NET produces an error stating that the name `Class1` is ambiguous.

## Namespace Level Statements

Within a namespace, you can define items such as modules, interfaces, classes, delegates, enumerations, structures, and other namespaces. You cannot define items such as properties, procedures, variables and events at the namespace level, these items must be declared within containers such as modules, structures, or classes.



# Multithreaded Applications

With Visual Basic .NET, you can write applications that perform multiple tasks simultaneously. Tasks with the potential of holding up other tasks can execute on separate threads, a process known as *multithreading* or *free threading*. Applications that use multithreading are more responsive to user input because the user interface stays active while processor-intensive tasks execute on separate threads. Multithreading is also useful when creating scalable applications, because you can add threads as the workload increases.

## Creating and Using Threads

You create a new thread in Visual Basic .NET by declaring a variable of type **System.Threading.Thread** and calling the constructor with the **AddressOf** statement and the name of the procedure or method you want to execute on the new thread. The following code provides an example:

```
Dim MyThread As New System.Threading.Thread(AddressOf MySub)
```

## Starting and Stopping Threads

To start the execution of a new thread, use the **Start** method, as in the following code:

```
MyThread.Start()
```

To stop the execution of a thread, use the **Abort** method, as in the following code:

```
MyThread.Abort()
```

Besides starting and stopping threads, you can also pause threads by calling the **Sleep** or **Suspend** methods, resume a suspended thread with the **Resume** method, and destroy a thread using the **Abort** method, as in the following code:

```
MyThread.Sleep ()
```

```
MyThread.Suspend ()
```



`MyThread.Abort ()`

See [Thread States <vaconthreadstates.htm>](#) for more information on thread states and methods.

## Thread Priorities

Every thread has a priority property — that determines how large of a time slice it gets to execute. The operating system allocates longer time slices to high-priority threads than it does to low-priority threads. New threads are created with the value of **Normal**, but you can adjust the **Priority** property to any of the other values in the **System.Threading.ThreadPriority**

enumeration.

See [ThreadPriority Enumeration](#) for a detailed description of the various thread priorities.

## Foreground and Background Threads

A *foreground thread* runs indefinitely, while a *background thread* terminates once the last foreground thread has stopped. You can use the **IsBackground** property to determine or change the background status of a thread

## Changing Thread States

Once a thread has started, you can call its methods to change its state. For example, you can cause a thread to pause for a fixed number of milliseconds by calling **Thread.Sleep**. The **Sleep** method takes as a parameter a timeout, which is the number of milliseconds that the thread remains blocked. Calling **Thread.Sleep(System.Threading.Timeout.Infinite)** causes a thread to sleep until it is interrupted by another thread that calls **Thread.Interrupt**. The **Thread.Interrupt** method wakes the destination thread out of any wait it may be in and causes an exception to be raised.

You can also pause a thread by calling **Thread.Suspend**. When a thread calls **Thread.Suspend** on itself, the call blocks until another thread resumes it by calling **Thread.Resume**. When a thread calls **Thread.Suspend** on another thread, the call is





non-blocking and causes the other thread to pause. Calling **Thread.Resume** breaks another thread out of its suspended state and causes it to resume execution. Unlike **Thread.Sleep**, **Thread.Suspend** does not immediately stop a thread; the suspended thread does not pause until the common language runtime determines that it has reached a safe point.

The **Thread.Abort** method stops a running thread by raising a **ThreadAbortException** exception that causes the thread to die.

See Thread Methods for detailed information about these methods

### Overview of ADO.NET

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed via OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

ADO.NET cleanly factors data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET **DataSet** object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remoted between tiers. The ADO.NET **DataSet** object can also be used independently of a .NET data provider to manage data local to the application or sourced from XML.

The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the **System.Data** namespace, reference both System.Data.dll and System.Xml.dll. For an example of compiling an ADO.NET application using a command line compiler, see ADO.NET Sample Application <cpconsampleapplication.htm>.

ADO.NET provides functionality to developers writing managed code similar to the functionality provided to native COM developers by ADO. For a discussion of the differences between ADO and ADO.NET, see "ADO.NET for the ADO Programmer" at <http://msdn.microsoft.com/library/en-us/dndotnet/html/ADONETProg.asp>.



## Design Goals for ADO.NET

As application development has evolved, new applications have become loosely coupled based on the Web application model. More and more of today's applications use XML to encode data to be passed over network connections. Web applications use HTTP as the fabric for communication between tiers, and therefore must explicitly handle maintaining state between requests. This new model is very different from the connected, tightly coupled style of programming that characterized the client/server era, where a connection was held open for the duration of the program's lifetime and no special handling of state was required.

In designing tools and technologies to meet the needs of today's developer, Microsoft recognized that an entirely new programming model for data access was needed, one that is built upon the .NET Framework. Building on the .NET Framework ensures that the data access technology would be uniform — components would share a common type system, design patterns, and naming conventions.

ADO.NET was designed to meet the needs of this new programming model: disconnected data architecture, tight integration with XML, common data representation with the ability to combine data from multiple and varied data sources, and optimized facilities for interacting with a database, all native to the .NET Framework.

In creating ADO.NET, Microsoft embraced the following design goals.

### **Leverage Current ADO Knowledge**

The design for ADO.NET addresses many of the requirements of today's application development model. At the same time, the programming model stays as similar as possible to ADO, so current ADO developers do not have to start from the beginning in learning a brand new data access technology. ADO.NET is an intrinsic part of the .NET Framework without seeming completely foreign to the ADO programmer.

ADO.NET coexists with ADO. While most new .NET-based applications will be written using ADO.NET, ADO remains available to the .NET programmer through .NET COM interoperability services.

For a discussion of the differences between ADO and ADO.NET, see "ADO.NET for the



ADO Programmer" at <http://msdn.microsoft.com/library/en-us/dndotnet/html/ADONETProg.asp>.

## **Support the N-Tier Programming Model**

ADO.NET provides first-class support for the disconnected, n-tier programming environment for which many new applications are written. The concept of working with a disconnected set of data has become a focal point in the programming model. The ADO.NET solution for n-tier programming is the **DataSet**.

## **Integrate XML Support**

XML and data access are intimately tied — XML is all about encoding data, and data access is increasingly becoming all about XML. The .NET Framework does not just support Web standards — it is built entirely on top of them.

XML support is built into ADO.NET at a very fundamental level. The XML classes in the .NET Framework and ADO.NET are part of the same architecture — they integrate at many different levels. You no longer have to choose between the data access set of services and their XML counterparts; the ability to cross over from one to the other is inherent in the design of both.

## **ADO.NET Architecture**

Data processing has traditionally relied primarily on a connection-based, two-tier model. As data processing increasingly uses multi-tier architectures, programmers are switching to a disconnected approach to provide better scalability for their applications.



## XML and ADO.NET

ADO.NET leverages the power of XML to provide disconnected access to data. ADO.NET was designed hand-in-hand with the XML classes in the .NET Framework — both are components of a single architecture.

ADO.NET and the XML classes in the .NET Framework converge in the **DataSet** object. The **DataSet** can be populated with data from an XML source, whether it is a file or an XML stream. The **DataSet** can be written as World Wide Web Consortium (W3C) compliant XML, including its schema as XML Schema definition language (XSD) schema, regardless of the source of the data in the **DataSet**. Because the native serialization format of the **DataSet** is XML, it is an excellent medium for moving data between tiers making the **DataSet** an optimal choice for remoting data and schema context to and from an XML Web service.

The **DataSet** can also be synchronized with an **XmlDataDocument** to provide relational and hierarchical access to data in real time. For more information, see [Synchronizing a DataSet with an XmlDataDocument](#)

[<cpconsynchronizingdatasetwithxmladatadocument.htm>](#).

## ADO.NET Components

The ADO.NET components have been designed to factor data access from data manipulation. There are two central components of ADO.NET that accomplish this: the **DataSet**, and the .NET data provider, which is a set of components including the **Connection**, **Command**, **DataReader**, and **DataAdapter** objects.

The ADO.NET DataSet [<cpcontheadonetdataset.htm>](#) is the core component of the disconnected architecture of ADO.NET. The **DataSet** is explicitly designed for data access independent of any data source. As a result it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application. The **DataSet**

contains a collection of one or more **DataTable** objects made up of rows and columns of data, as well as primary key, foreign key, constraint, and relation information about the data in the **DataTable** objects.



## Mac Id Authentication For HR Employee Information

---

The other core element of the ADO.NET architecture is the .NET data provider <cpconadonetproviders.htm>, whose components are explicitly designed for data manipulation and fast, forward-only, read-only access to data. The **Connection** object provides connectivity to a data source. The **Command** object enables access to database commands to return data, modify data, run stored procedures, and send or retrieve parameter information. The **DataReader** provides a high-performance stream of data from the data source. Finally, the **DataAdapter** provides the bridge between the **DataSet** object and the data source. The **DataAdapter** uses **Command** objects to execute SQL commands at the data source to both load the **DataSet** with data, and reconcile changes made to the data in the **DataSet** back to the data source.

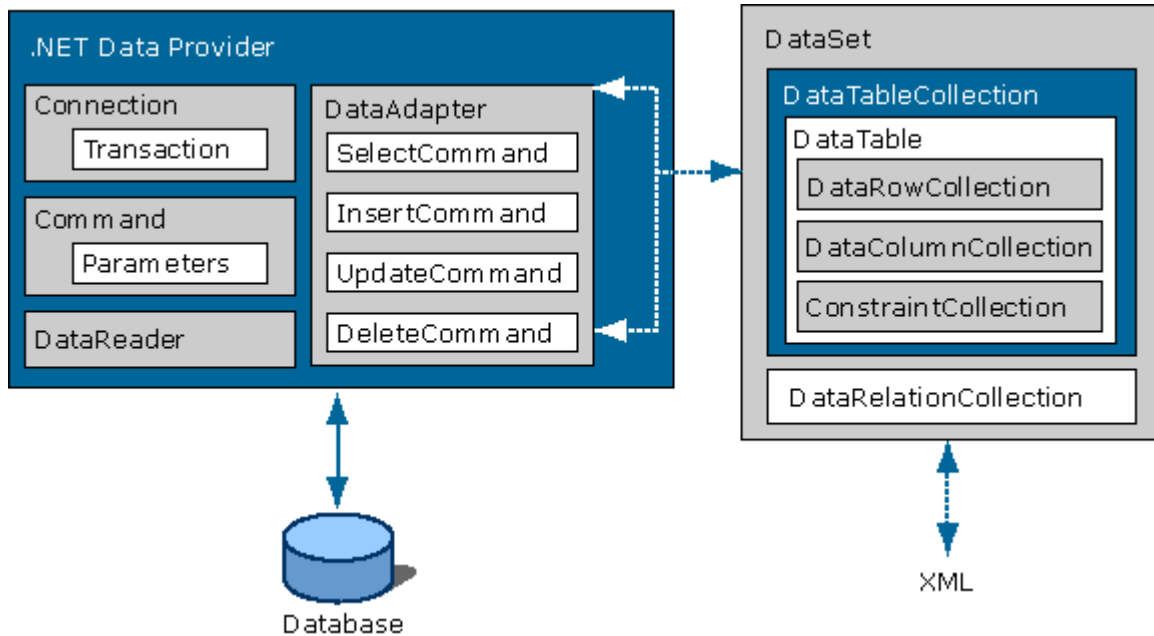
You can write .NET data providers for any data source. The .NET Framework ships with two .NET data providers

the SQL Server .NET Data Provider and the OLE DB .NET Data Provider.

The following diagram illustrates the components of ADO.NET architecture the SQL Server .NET Data Provider and the OLE DB .NET Data Provider.

The following diagram illustrates the components of ADO.NET architecture

### **ADO.NET architecture**



## Remoting or Marshaling Data between Tiers and Clients

The design of the **DataSet** enables you to easily transport data to clients over the Web using XML Web services, as well as allowing you to marshal data between .NET components using .NET Remoting services. You can also remote a strongly typed **DataSet** in this fashion. For an overview of XML Web services, see [XML Web Services Overview <cpconwebservicesservicesoverview.htm>](#). For an example of consuming a **DataSet** from an XML Web service, see [Consuming a DataSet from an XML Web Service <cpconconsumingdatasetfromwebservice.htm>](#).

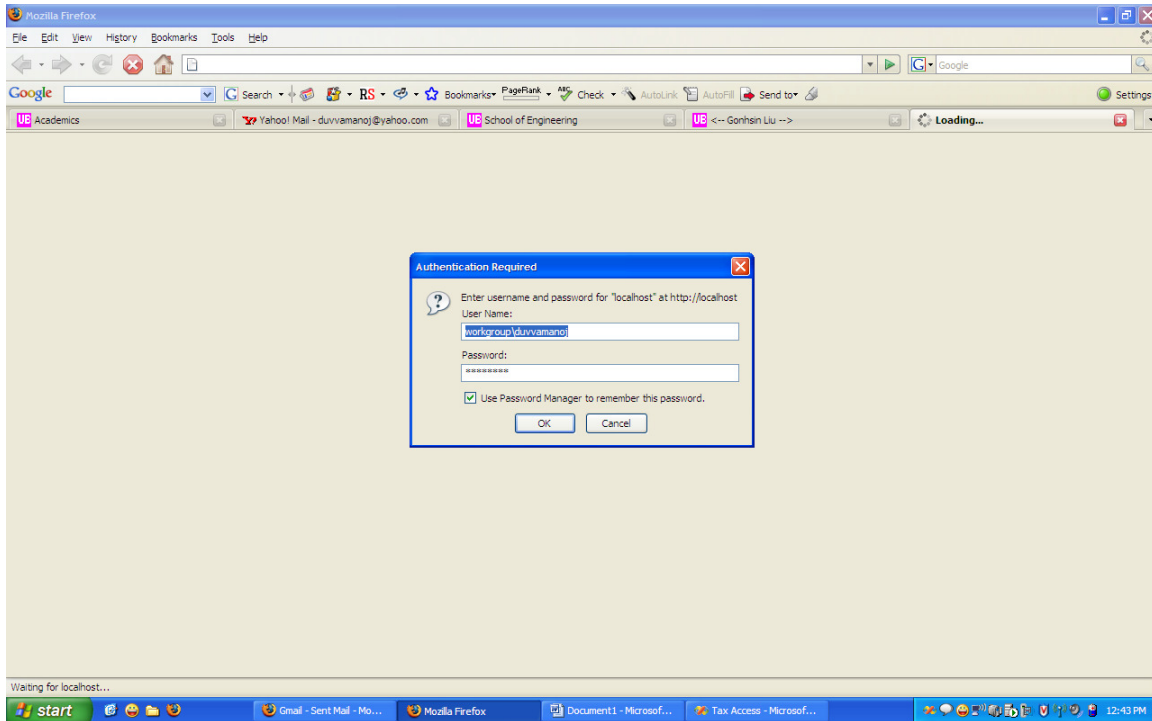
An overview of remoting services can be found in the [.NET Remoting Overview <cpconnetremotingoverview.htm>](#). Note that **DataTable** objects can also be used with remoting services, but cannot be transported via an XML Web service.

## ADO.NET Platform Requirements

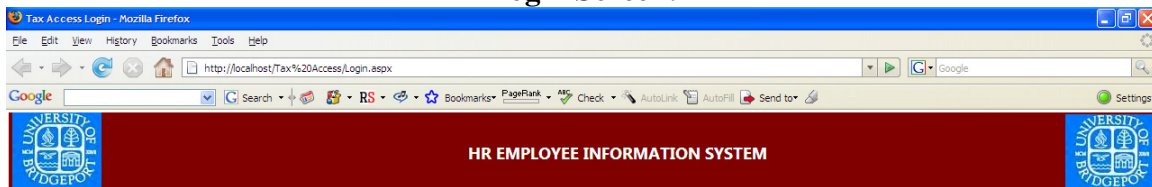
The Microsoft .NET Framework SDK (including ADO.NET) is supported on Microsoft® Windows® 2000, Microsoft® Windows NT® 4 with Service Pack 6a, Microsoft® Windows® Millennium Edition, Microsoft® Windows® 98, and Microsoft® Windows® SE. Use of the SQL Server .NET Data Provider or OLE DB .NET Data Provider requires the installation of Microsoft Data Access Components version 2.6 or later.

## SCREENS

### Authentication screen:

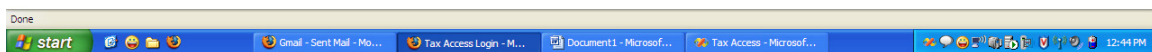


### Login Screen:



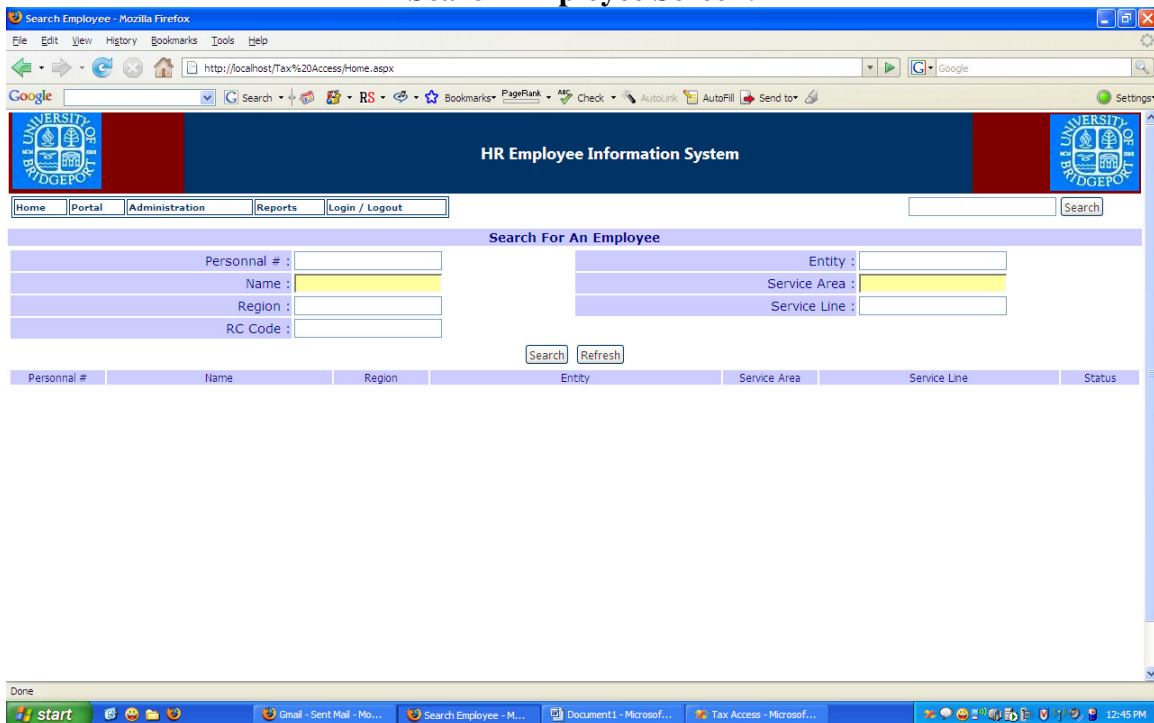
Login Id :

Password :



## Mac Id Authentication For HR Employee Information

### Search Employee Screen:



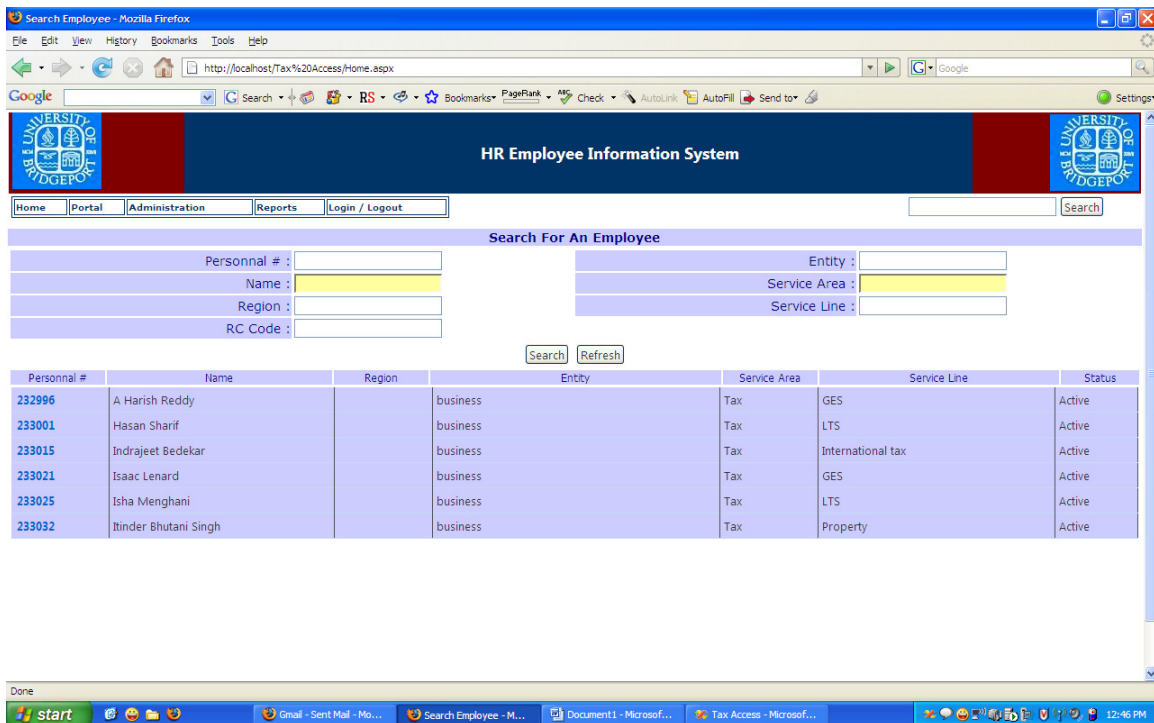
**HR Employee Information System**

Home | Portal | Administration | Reports | Login / Logout

**Search For An Employee**

Personal # :  Entity :   
 Name :  Service Area :   
 Region :  Service Line :   
 RC Code :

Personal #	Name	Region	Entity	Service Area	Service Line	Status



**HR Employee Information System**

Home | Portal | Administration | Reports | Login / Logout

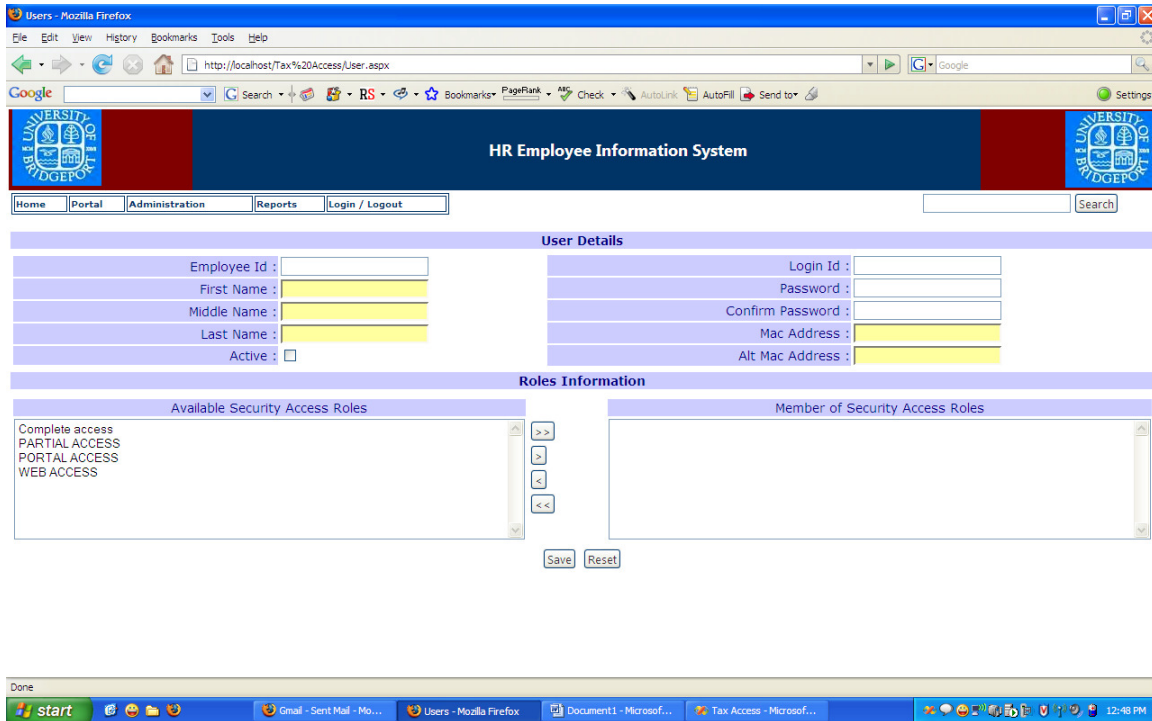
**Search For An Employee**

Personal # :  Entity :   
 Name :  Service Area :   
 Region :  Service Line :   
 RC Code :

Personal #	Name	Region	Entity	Service Area	Service Line	Status
232996	A Harish Reddy		business	Tax	GES	Active
233001	Hasan Sharif		business	Tax	LTS	Active
233015	Indrajeet Bedekar		business	Tax	International tax	Active
233021	Isaac Lenard		business	Tax	GES	Active
233025	Isha Menghani		business	Tax	LTS	Active
233032	Itinder Bhutani Singh		business	Tax	Property	Active



## Administration Add new user:



**HR Employee Information System**

Home | Portal | **Administration** | Reports | Login / Logout

**User Details**

Employee Id :   
 First Name :   
 Middle Name :   
 Last Name :   
 Active :

Login Id :   
 Password :   
 Confirm Password :   
 Mac Address :   
 Alt Mac Address :

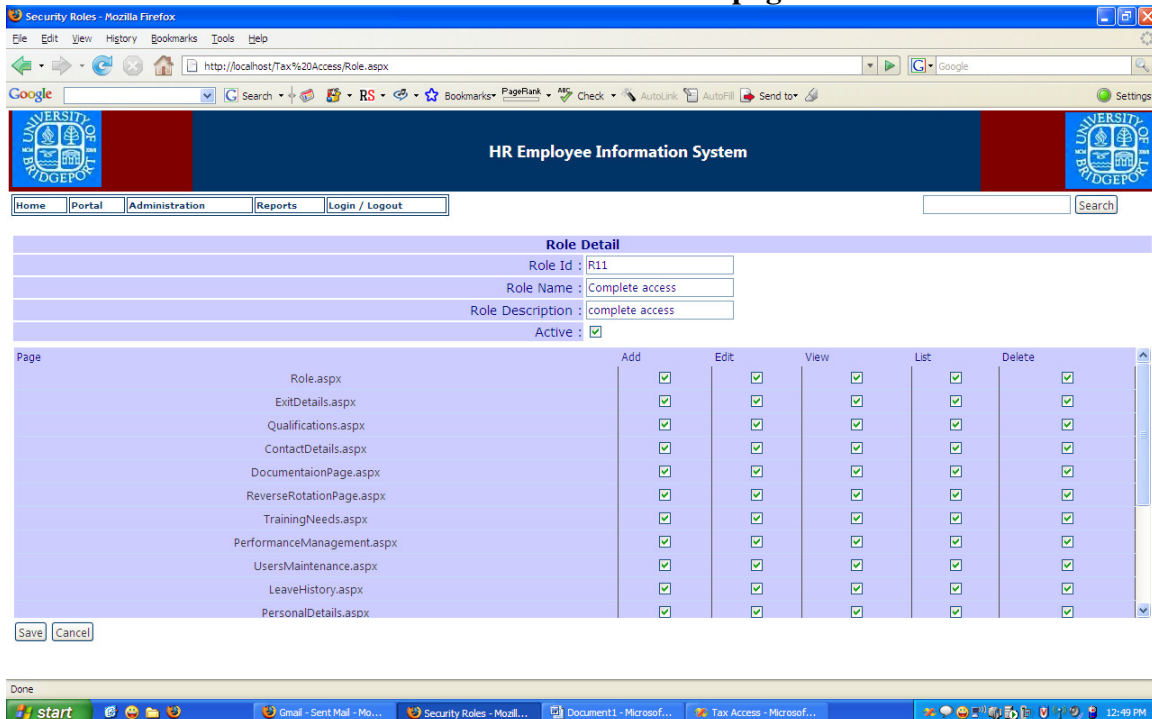
**Roles Information**

Available Security Access Roles: Complete access, PARTIAL ACCESS, PORTAL ACCESS, WEB ACCESS

Member of Security Access Roles:

Save Reset

## Administration Role detail page:



**HR Employee Information System**

Home | Portal | **Administration** | Reports | Login / Logout

**Role Detail**

Role Id : R11  
 Role Name : Complete access  
 Role Description : complete access  
 Active :

Page	Add	Edit	View	List	Delete
Role.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ExitDetails.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Qualifications.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ContactDetails.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DocumentaionPage.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ReverseRotationPage.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TrainingNeeds.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PerformanceManagement.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
UsersMaintenance.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
LeaveHistory.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PersonalDetails.aspx	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel



## Mac Id Authentication For HR Employee Information

### Employee Information page:

The screenshot shows the 'Employee Information' page in a Mozilla Firefox browser. The page title is 'HR Employee Information System'. It features a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. The main content area is divided into two sections: 'Employee Details' and 'FSS DETAILS'. The 'Employee Details' section contains a grid of input fields for various employee attributes. The 'FSS DETAILS' section is currently empty. A red note at the bottom of the 'Employee Details' section states: '\* Note: Please make sure the date you enter is in MM/DD/YYYY format'. The browser's taskbar at the bottom shows several open applications, including 'Gmail - Sent Mail - Mo...', 'Employee Informatio...', 'Document1 - Microsof...', and 'Tax Access - Microsof...'. The system clock shows 12:50 PM.

Employee Details		
Empl Code :	File No. :	Cupbopard No. :
DOB :	DOJ :	Desg :
Desg Code :	Legal Entity :	RC Code :
RC Short Name :	Role :	Location :
Function :	Service Area :	Service Line :
Region :	R10 Team Lead :	US Team Lead :
Co-Ordinator :	HR Generalist :	Qualification :
Year Passing :	Relevant Experience :	Deloitte Experience :
Total Experience :	Official E-Mail :	FSS Move Date :

\* Note: Please make sure the date you enter is in MM/DD/YYYY format

FSS DETAILS	
-------------	--

### Employee Experience information:

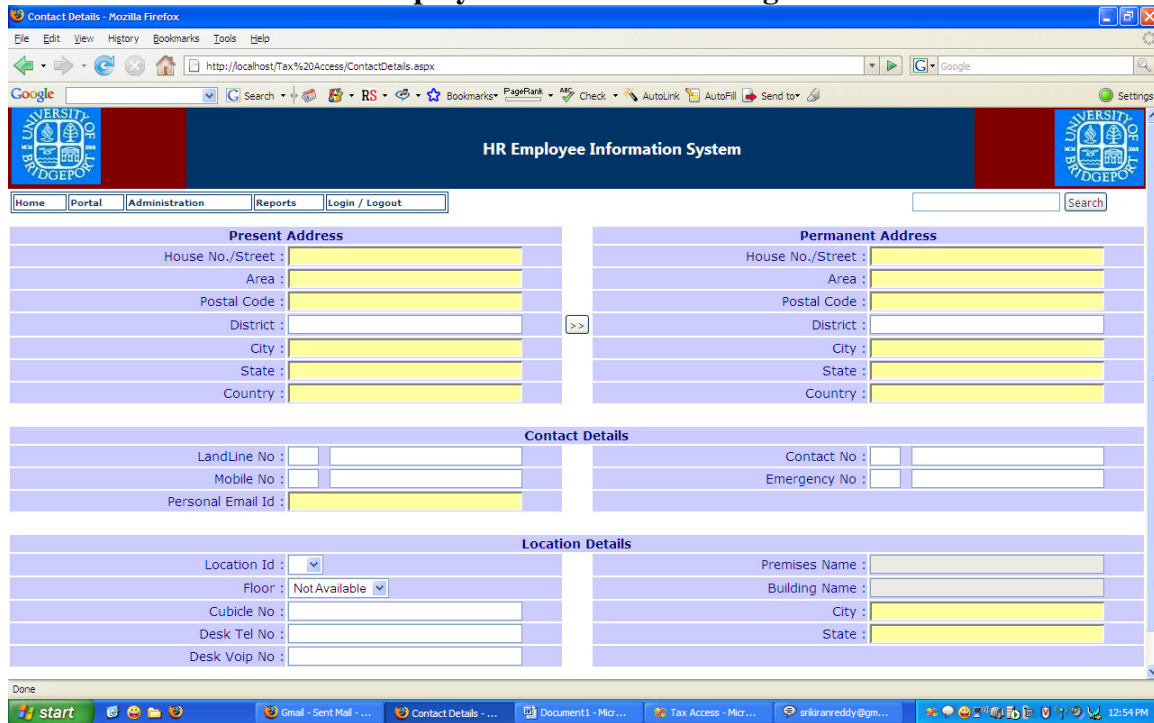
The screenshot shows the 'Experience Details' page in a Mozilla Firefox browser. The page title is 'HR Employee Information System'. It features a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. The main content area is divided into two sections: 'Experience Details' and 'Reference Details'. The 'Experience Details' section contains a table with columns for 'Action', 'Company Name', 'Location', 'Employer Industry', 'Job Title', 'From Date', 'To Date', and 'Exp.(in Mts)'. Below the table, there is a summary box showing 'Total Experience In Months' as 0. The 'Reference Details' section contains a 'Reference Check' checkbox and a 'Report/Document' dropdown menu. A red note at the top of the 'Experience Details' section states: '\* Note: Please make sure the date you enter is in MM/DD/YYYY format'. The browser's taskbar at the bottom shows several open applications, including 'Gmail - Sent Mail - Mo...', 'Experience Details - ...', 'Document1 - Microsof...', and 'Tax Access - Microsof...'. The system clock shows 12:51 PM.

Experience Details							
Action	Company Name	Location	Employer Industry	Job Title	From Date	To Date	Exp.(in Mts)

Total Experience In Months: 0

Reference Details	
<input type="checkbox"/> Reference Check	
Report/Document	

### Employee Contact Details Page:



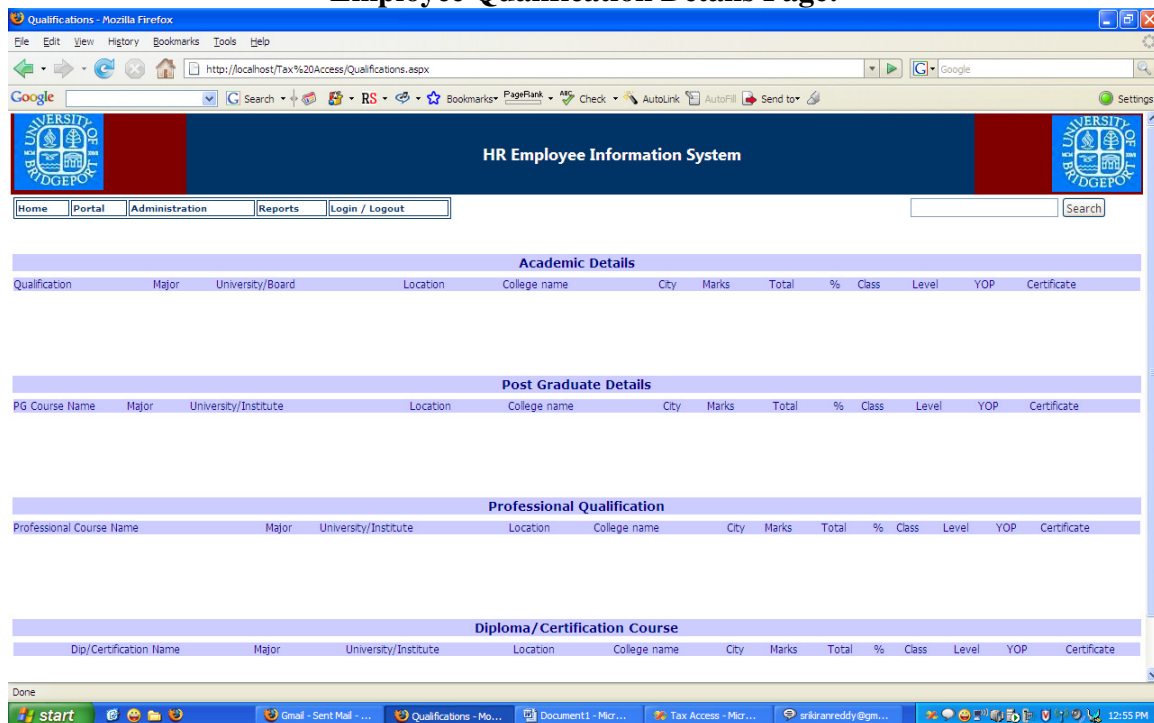
The screenshot shows a web browser window displaying the 'HR Employee Information System'. The page title is 'Contact Details - Mozilla Firefox'. The URL is 'http://localhost/Tax%20Access/ContactDetails.aspx'. The page features a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. A search bar is located on the right side of the navigation menu.

The main content area is divided into several sections:

- Present Address:** Fields for House No./Street, Area, Postal Code, District, City, State, and Country.
- Permanent Address:** Fields for House No./Street, Area, Postal Code, District, City, State, and Country.
- Contact Details:** Fields for LandLine No., Mobile No., Personal Email Id, Contact No., and Emergency No.
- Location Details:** Fields for Location Id (dropdown), Floor (dropdown), Cubicle No., Desk Tel No., Desk Voip No., Premises Name, Building Name, City, and State.

The browser's taskbar at the bottom shows several open applications: 'start', 'Gmail - Sent Mail - ...', 'Contact Details - ...', 'Document1 - Micr...', 'Tax Access - Micr...', and 'srikiranreddy@sgm...'. The system clock shows 12:54 PM.

### Employee Qualification Details Page:



The screenshot shows a web browser window displaying the 'HR Employee Information System'. The page title is 'Qualifications - Mozilla Firefox'. The URL is 'http://localhost/Tax%20Access/Qualifications.aspx'. The page features a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. A search bar is located on the right side of the navigation menu.

The main content area is divided into several sections, each with a table structure:

- Academic Details:** Table with columns: Qualification, Major, University/Board, Location, College name, City, Marks, Total, % Class, Level, YOP, Certificate.
- Post Graduate Details:** Table with columns: PG Course Name, Major, University/Institute, Location, College name, City, Marks, Total, % Class, Level, YOP, Certificate.
- Professional Qualification:** Table with columns: Professional Course Name, Major, University/Institute, Location, College name, City, Marks, Total, % Class, Level, YOP, Certificate.
- Diploma/Certification Course:** Table with columns: Dip/Certification Name, Major, University/Institute, Location, College name, City, Marks, Total, % Class, Level, YOP, Certificate.

The browser's taskbar at the bottom shows several open applications: 'start', 'Gmail - Sent Mail - ...', 'Qualifications - Mo...', 'Document1 - Micr...', 'Tax Access - Micr...', and 'srikiranreddy@sgm...'. The system clock shows 12:55 PM.



## Mac Id Authentication For HR Employee Information

### Employee Personal Details Page:

Personal Details - Mozilla Firefox  
http://localhost/Tax%20Access/PersonalDetails.aspx

HR Employee Information System

Home Portal Administration Reports Login / Logout

\* Note: Please make sure the date you enter is in MM/DD/YYYY format

**Family Details**

Father's Name :  Date Of Marriage :   
Mother's Name :  Marital Status :   
Spouse Name :

**Child Details**

Action	Child Name	Gender	Age	Date Of Birth
--------	------------	--------	-----	---------------

**Document Details**

Passport No :  Pan No :   
Place Of Issue :  Copy Enclosed :   
Issue Date :  Blood Group :   
Expiry Date :  Training Bond :   
Copy Enclosed :

**Other Details**

Source Of DOB :  Visa Document :

Done

start Gmail - Sent Mail - ... Personal Details - ... Document1 - Micr... Tax Access - Micr... srikanthv8@gmail... 1:00 PM

srikanthv8  
andhulo chusi trisha place ni  
mardhalu laga fell  
avuthunnademiti

### Employee Exit Details:

Exit Details - Mozilla Firefox  
http://localhost/Tax%20Access/ExitDetails.aspx

HR Employee Information System

Home Portal Administration Reports Login / Logout

**Exit Details**

End Date :  Exit Reason For Leaving :   
Exit Reason Code :  Exit Attrition Type :   
Exit Spot Rating :  Exit Region :   
Exit SA :  Exit SL :

Reset

Done

start Gmail - Sent Mail - ... Exit Details - Mozil... Document1 - Micr... Tax Access - Micr... srikanthv8@gmail... 1:02 PM



## Mac Id Authentication For HR Employee Information

### Employee Performance Management Page:

The screenshot shows the 'Performance Management' page of the HR Employee Information System. The browser window title is 'Performance Management - Mozilla Firefox'. The address bar shows the URL: `http://localhost/Tax%20Access/PerformanceManagement.aspx`. The page header includes the University of Bridgport logo and the text 'HR Employee Information System'. Below the header is a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. The main content area is titled 'Current Year Details' and contains several dropdown menus for 'MidYear Forms', 'YearEnd Forms', 'PDP Forms', 'MidYear Status', and 'YearEnd Status'. A 'Reset' button is located below these forms. A note states: '\* Note: Please make sure the date you enter is in MM/DD/YYYY format'. Below this is a table titled 'Counsellors History' with columns for 'Action', 'Counsellors Name', 'Start Date', 'End Date', and 'Remarks'. The table is currently empty. Another table titled 'Feedback' has columns for 'Action', 'FeedBack Rating', 'Provider', 'Engagement', and 'Feedback Date', also currently empty. The browser's taskbar at the bottom shows the 'start' button, two Firefox windows, and several other applications.

### Employee Report Page:

The screenshot shows the 'Reports' page of the HR Employee Information System. The browser window title is 'Reports - Mozilla Firefox'. The address bar shows the URL: `http://localhost/Tax%20Access/Reports.aspx?Rep_Id=Employee%20Details`. The page header includes the University of Bridgport logo and the text 'HR Employee Information System'. Below the header is a navigation menu with 'Home', 'Portal', 'Administration', 'Reports', and 'Login / Logout'. The main content area is titled 'Reports' and contains several dropdown menus for 'Entity', 'Service Line', 'RC-Code', 'Designation', 'Date Of Joining', 'Location', and 'Personnel No'. There are also text input fields for 'Date Of Joining' and 'Personnel No'. 'Generate Report' and 'Export' buttons are located at the bottom of the form. The browser's taskbar at the bottom shows the 'start' button, a Gmail window, and several other applications.

**DATABASE TABLES:****Amain:**

Personnel_No	nvarchar	10	1
GUID#	nvarchar	10	1
[Eml_Code]	nvarchar	5	1
File_no	nvarchar	10	1
Cupboard_No	nvarchar	12	1
Audit_Status	nvarchar	16	1
Gender	nvarchar	3	1
Name	nvarchar	60	1
DOB	smalldatetime	4	1
DOJ	smalldatetime	4	1
Desg_code	smallint	2	1
Desg	nvarchar	60	1
Discretionary_Title	nvarchar	65	1
Lega_entity	nvarchar	50	1
RC_Code	nvarchar	12	1
[RC_short_name]	nvarchar	50	1
Role	nvarchar	50	1
Location	nvarchar	15	1
[Practice_function]	nvarchar	40	1
[Service_Area]	nvarchar	40	1
[Service_Line]	nvarchar	40	1
Region	nvarchar	20	1
R10_Team_Lead	nvarchar	50	1
US_Team_Lead	nvarchar	50	1
[Co-Ordinator]	nvarchar	50	1
HR_Generalist	nvarchar	50	1
Qualification	nvarchar	65	1
Year_Passing	nvarchar	25	1
[Relavant_Experience]	smallint	2	1
[Deloitte_Experience]	smallint	2	1
[Total_Experience]	smallint	2	1
[Official_email]	nvarchar	50	1
Old_Title_Code	nvarchar	2	1
Old_Title	nvarchar	60	1



## Mac Id Authentication For HR Employee Information

Employee_Status	nvarchar	15	1
DT_Seperated	smalldatetime	4	1
Father_Name	nvarchar	60	1
Mother_Name	Nvarchar	60	1
marital_status	nvarchar	12	1
Date_Marriage	smalldatetime	4	1
Spouse_name	nvarchar	55	1
Passport_No	nvarchar	18	1
[Passport-place]	nvarchar	35	1
Passport_issue	smalldatetime	4	1
[Passport-expiry]	smalldatetime	4	1
[Passport-enclosed]	nvarchar	20	1
PAN_No	nvarchar	15	1
PAN_Enclosed	nvarchar	20	1
Blood_Group	nvarchar	6	1
Present_Add_1	nvarchar	80	1
Present_Add_2	nvarchar	80	1
Present_Add_3	nvarchar	35	1
Present_Add_4	nvarchar	15	1
Present_Add_5	nvarchar	25	1
Present_Add_6	nvarchar	30	1
Present_Add_7	nvarchar	15	1
[Phone_land]	nvarchar	35	1
[Phone_hand]	nvarchar	20	1
Emergency_PhoneNo	nvarchar	45	1
[Personal_email]	nvarchar	60	1
Perm_Hous_Street	nvarchar	230	1
Perm_Area	nvarchar	100	1
Perm_Dist	nvarchar	50	1
Perm_Postal_Code	nvarchar	50	1
Permt_City	nvarchar	50	1
Permt_State	nvarchar	50	1
Permt_Country	nvarchar	50	1
Permanent_Add_ContactNo	nvarchar	50	1
Qual1	nvarchar	75	1
Qual1_Major	nvarchar	50	1



## Mac Id Authentication For HR Employee Information

Qual1_University_Board	nvarchar	90	1
Qual1_Location	nvarchar	20	1
Qual1_College_Name	nvarchar	90	1
Qual1_College_City	nvarchar	20	1
Qual1_Marks_Obtained	nvarchar	4	1
Qual1_Total_Marks	nvarchar	4	1
Qual1_Percentage	decimal	9	1
Qual1_Class_Division	nvarchar	20	1
Qual1_Level	nvarchar	20	1
Qual1_YOP	smalldatetime	4	1
Qual1Cert_Enclosed	nvarchar	20	1
Qual2	nvarchar	75	1
Qual2_University_Board	nvarchar	75	1
Qual2_Location	nvarchar	40	1
Qual2_College_Name	nvarchar	90	1
Qual2_College_City	nvarchar	20	1
Qual2_Marks_Obtained	nvarchar	4	1
Qual2_Total_Marks	nvarchar	4	1
Qual2_Percentage	decimal	9	1
Qual2_Class_Division	nvarchar	20	1
Qual2_Level	nvarchar	20	1
Qual2_YOP	smalldatetime	4	1
Qual2Cert_Enclosed	nvarchar	20	1
Qual3	nvarchar	75	1
Qual3_University_Board	nvarchar	75	1
Qual3_Location	nvarchar	20	1
Qual3_College_Name	nvarchar	90	1
Qual3_College_City	nvarchar	20	1
Qual3_Marks_Obtained	nvarchar	4	1
Qual3_Total_Marks	nvarchar	4	1
Qual3_Percentage	decimal	9	1
Qual3_Class_Division	nvarchar	20	1
Qual3_Level	nvarchar	20	1
Qual3_YOP	smalldatetime	4	1
Qual3Cert_Enclosed	nvarchar	20	1





Mac Id Authentication For HR Employee Information

Reference_Chk	bit	1	0
Reference_copy	nvarchar	12	1

**ADMINCOPY**

Personnel_No	int	4	0
GUID#	int	4	1
[Eml_Code]	numeric	9	1
File_no	varchar	10	1
Cupboard_No	varchar	12	1
Audit_Status	varchar	16	1
Gender	char	3	1
Name	varchar	60	1
DOB	datetime	8	1
DOJ	datetime	8	1
Desg_code	int	4	1
Desg	varchar	60	1
Discretionary_Title	varchar	65	1
Lega_entity	varchar	50	1
RC_Code	varchar	12	1
[RC_short_name]	varchar	50	1
Role	varchar	50	1
Location	varchar	15	1
[Practice_function]	varchar	40	1
[Service_Area]	varchar	40	1
[Service_Line]	varchar	40	1
Region	varchar	20	1
R10_Team_Lead	varchar	20	1
US_Team_Lead	varchar	50	1
[Co-Ordinator]	varchar	50	1
HR_Generalist	varchar	50	1
Qualification	varchar	65	1
Year_Passing	varchar	50	1
[Relavant_Experience]	int	4	1
[Deloitte_Experience]	int	4	1
[Total_Experience]	int	4	1
[Official_email]	varchar	50	1
Old_Title_Code	varchar	2	1



## Mac Id Authentication For HR Employee Information

Old_Title	varchar	60	1
Employee_Status	varchar	15	1
DT_Seperated	datetime	8	1
Father_Name	varchar	60	1
Mother_Name	varchar	60	1
marital_status	varchar	12	1
Date_Marriage	datetime	8	1
Spouse_name	varchar	55	1
Passport_No	varchar	18	1
[Passport-place]	varchar	35	1
Passport_issue	datetime	8	1
[Passport-expiry]	datetime	8	1
[Passport-enclosed]	varchar	20	1
PAN_No	varchar	15	1
PAN_Enclosed	varchar	20	1
Blood_Group	varchar	6	1
Present_Add_1	varchar	80	1
Present_Add_2	varchar	80	1
Present_Add_3	varchar	35	1
Present_Add_4	varchar	15	1
Present_Add_5	varchar	25	1
Present_Add_6	varchar	30	1
Present_Add_7	varchar	15	1
[Phone_land]	varchar	35	1
[Phone_hand]	varchar	20	1
Emergency_PhoneNo	varchar	45	1
[Personal_email]	varchar	60	1
Perm_Hous_Street	varchar	230	1
Perm_Area	varchar	100	1
Perm_Dist	varchar	50	1
Perm_Postal_Code	varchar	50	1
Permt_City	varchar	50	1
Permt_State	varchar	50	1
Permt_Country	varchar	50	1
Permanent_Add_ContactNo	varchar	50	1
Qual1	varchar	75	1
Qual1_Major	varchar	50	1
Qual1_University_Board	varchar	90	1



Mac Id Authentication For HR Employee Information

Qual1_Location	nvarchar	20	1
Qual1_College_Name	nvarchar	90	1
Qual1_College_City	nvarchar	20	1
Qual1_Marks_Obtained	nvarchar	4	1
Qual1_Total_Marks	nvarchar	4	1
Qual1_Percentage	decimal	9	1
Qual1_Class_Division	nvarchar	20	1
Qual1_Level	nvarchar	20	1
Qual1_YOP	smalldatetime	4	1
Qual1Cert_Enclosed	nvarchar	20	1
Qual2	nvarchar	75	1
Qual2_University_Board	nvarchar	75	1
Qual2_Location	nvarchar	40	1
Qual2_College_Name	nvarchar	90	1
Qual2_College_City	nvarchar	20	1
Qual2_Marks_Obtained	nvarchar	4	1
Qual2_Total_Marks	nvarchar	4	1
Qual2_Percentage	decimal	9	1
Qual2_Class_Division	nvarchar	20	1
Qual2_Level	nvarchar	20	1
Qual2_YOP	smalldatetime	4	1
Qual2Cert_Enclosed	nvarchar	20	1
Qual3	nvarchar	75	1
Qual3_University_Board	nvarchar	75	1
Qual3_Location	nvarchar	20	1
Qual3_College_Name	nvarchar	90	1
Qual3_College_City	nvarchar	20	1
Qual3_Marks_Obtained	nvarchar	4	1
Qual3_Total_Marks	nvarchar	4	1
Qual3_Percentage	decimal	9	1
Qual3_Class_Division	nvarchar	20	1
Qual3_Level	nvarchar	20	1
Qual3_YOP	smalldatetime	4	1
Qual3Cert_Enclosed	nvarchar	20	1
Reference_Chk	bit	1	0
Reference_copy	nvarchar	12	1
Backgroud_Verif	bit	1	0
Backgroud_copy	nvarchar	12	1
Backgroud_Code	nvarchar	25	1
Sourcedata_DOB	nvarchar	30	1



## Mac Id Authentication For HR Employee Information

Sports	nvarchar	100	1
Extracirculam_actyvt	nvarchar	120	1
photo	image	16	1
Joining_Form	nvarchar	20	1
[Offer_Letter]	nvarchar	20	1
Past_Relieving_Ltr	nvarchar	20	1
Past_PaySlip_SalCertific	nvarchar	20	1
Form_16	nvarchar	20	1
Resume	nvarchar	20	1
Interview_Evaluation	nvarchar	20	1
Independence_Compl	nvarchar	20	1
Old_GUID	nvarchar	15	1
FSS_move_date	nvarchar	15	1
Past_Entity	nvarchar	60	1
Visa	nvarchar	18	1
Visa_Document	nvarchar	18	1
Languages_Know	nvarchar	60	1
Trg_Bond	nvarchar	18	1
Joining_bonus	nvarchar	18	1
Relocation_All	nvarchar	20	1
Confirmation_Ltr	nvarchar	20	1
Documents_submitted	nvarchar	200	1
[tag for tax]	nvarchar	1	1

### Answer\_table:

Ans_Id	int	4	0
Q_Id	int	4	1
Ans_Text	varchar	100	1
Ans_Type	varchar	25	1

### Group\_table:

Group_Id	int	4	0
Group_Text	varchar	50	1
Category_Id	int	4	1

### Leave\_Details:

Id	int	4	0
Personnel_No	int	4	0



Leave_Start_Date	datetime	8	0
Leave_End_Date	datetime	8	1
Leave_Type	varchar500	1	
Leave_No	int	4	1

**Master\_Answer:**

Question_Id	int	4	1
Answer_Text	nvarchar	50	1
Answer_id	int	4	0
ControlType	nvarchar	50	1
Correct_Ans	nvarchar	50	1

**Master\_CareerGref:**

CareerGraf_Id	int	4	0
CareerGraf_Text	varchar	50	0
Description	varchar	100	1

**Master\_Category:**

Cat_Id	int	4	0
Cat_Name	nvarchar	50	1

**Master\_Design:**

Desg_Code	int	4	0
Designation	varchar	50	1

**Master\_EntityName**

Entity_ID	int	4	0
Entity_Name	varchar	50	0

**Master\_Groups:**

Grp_id	int	4	0
Grp_Name	nvarchar	50	1
[Cat_id (FK)]	int	4	1

**Master\_Location:**

Location_ID	int	4	0
Premises_Name	vvarchar	15	1
Building_Name	vvarchar	15	1
City	vvarchar	15	1
State	vvarchar	15	1

**Master\_Main:**

Personnel_No	int	4	0
GUID	int	4	1
Eml_Code	int	4	1
File_no	char	9	1
Cupboard_No	char	9	1
Audit_Status	vvarchar	500	1
Gender	char	20	1
Name	vvarchar	500	1
DOB	datetime	8	1
DOJ	datetime	8	1
Desg_code	int	4	1
Desg	vvarchar	500	1
Discretionary_Title	vvarchar	500	1
Entity_ID	int	4	1
RC_Code	vvarchar	500	1
RC_short_name	vvarchar	500	1
Role	vvarchar	500	1
Location	vvarchar	500	1
Practice_function	vvarchar	500	1
Service_Area	vvarchar	500	1
Service_Line	vvarchar	500	1
Region	vvarchar	500	1
R10_Team_Lead	vvarchar	500	1
US_Team_Lead	vvarchar500	1	
Co_Ordinator	vvarchar500	1	
HR_Generalist	vvarchar	500	1
Qualification	vvarchar	500	1



## Mac Id Authentication For HR Employee Information

Year_Passing	varchar	500	1
Relavant_Experience	int	4	1
Deloitte_Experience	int	4	1
Total_Experience	int	4	1
Official_email	varchar	500	1
Old_Title_Code	varchar	500	1
Old_Title	varchar	500	1
Employee_Status	varchar	500	1
DT_Seperated	datetime	8	1
Father_Name	varchar	500	1
Qual1_Class_Division	varchar	500	1
Qual1_Level	varchar	500	1
Qual1_YOP	varchar	500	1
Qual1Cert_Enclosed	varchar	500	1
Qual2	varchar	500	1
Qual2_University_Board	varchar	500	1
Qual2_Location	varchar	500	1
Qual2_College_Name	varchar	500	1
Qual2_College_City	varchar	500	1
Qual2_Marks_Obtained	varchar	500	1
Qual2_Total_Marks	varchar	500	1
Qual2_Percentage	decimal	9	1
Qual2_Class_Division	varchar	500	1
Qual2_Level	varchar	500	1
Qual2_YOP	datetime	8	1
Qual2Cert_Enclosed	varchar	500	1
Qual3	varchar	500	1
Qual3_University_Board	varchar	500	1
Qual3_Location	varchar	500	1

### Security\_Pages:

pkey_page	int	4	0
page_id	varchar	50	0
page_name	varchar	100	0
page_url	varchar	100	1
fkey_user	int	4	1
last_updated_date	datetime	8	0



### Security\_Role\_Page\_Xref:

pkey_xref	int	4	0
fkey_role	int	4	0
fkey_page	int	4	0
[add]	bit	1	0
edit	bit	1	0
[view]	bit	1	0
[delete]	bit	1	0
list	bit	1	0
fkey_user	int	4	1
last_updated_date	datetime8		0

### Security\_Roles:

pkey_role	int	4	0
role_id	varchar	50	
role_name	varchar	100	0
role_desc	varchar	300	1
is_active	bit	1	0
fkey_user	int	4	1
last_updated_date	datetime	8	0

### Security\_User\_Role\_Xref:

pkey_xref	int	4	0
fkey_role	int	4	0
user_key	int	4	0
fkey_user	int	4	1
last_updated_date	datetime8		





### Security\_Users:

pkey_user	int	4	0
first_name	varchar	50	0
middle_name	varchar	50	1
last_name	varchar	50	0
login_id	varchar	100	0
password	varchar	50	0
employee_id	varchar	50	0
canchangepassword	bit	1	1
is_active	bit	1	0
macaddress	varchar	100	0
Alt_macaddress	varchar	100	1
last_updated_date	datetime	8	0
Visibility	varchar	100	1

### Assignment\_Table:

ExitPersonnel_Id	int	4	0
GroupAssigned	int	4	0
Employee_Assigned	int	4	1

### Category\_Table:

Category_Id	int	4	0
Category_Text	varchar	100	1

### Child Master\_table:

Id	int	4	0
Child_report_Master	varchar	100	0
F_PK_Primary_Report_Master	int	4	0

### Details\_Acadimic:

pkey	int	4	0
personnel_no	int	4	0
qualification	varchar	75	0
major	varchar	50	1
University	varchar	90	1
Location	varchar	20	1



## Mac Id Authentication For HR Employee Information

College_Name	varchar	90	1
College_City	varchar	20	1
Marks_Obtained	varchar	4	1
Total_Marks	varchar	4	1
Percentage	varchar	20	1
Class	varchar	1	1
[Level]	varchar	50	1
YOP	varchar	10	1
certificate_Enclosed	bit	1	0
fkey_user	int	4	1

### Education:

ID	int	4	0
Pers_no	int	4	0
Course	varchar	50	1
Major	varchar	60	1
Institute	varchar	80	1
Location	varchar	30	1
College_Name	varchar	90	1
College_City	varchar	20	1
Marks_Obtained	varchar	4	1
Total_Marks	varchar	4	1
Percentage	decimal	9	1
Class_Division	varchar	20	1
[Level]	varchar	20	1
YOP	datetime	8	1
Enclosed	varchar	20	1

### Details\_Deploma\_Qualification:

pkey	int	4	0
Personnel_No	int	4	0
DC_Course	varchar	500	0
DC_Major	varchar	500	1
DC_Institute	varchar	500	1
DC_Location	varchar	500	1
DC_College_Name	varchar	500	1
DC_College_City	varchar	500	1



## Mac Id Authentication For HR Employee Information

DC_Marks_Obtained	varchar	500	1
DC_Total_Marks	varchar	500	1
DC_Percentage	varchar	500	1
DC_Class_Division	varchar	500	1
DC_Level	varchar	500	1
DC_YOP	varchar	500	1
DCCert_Enclosed	bit	1	0
fkey_user	int	4	1
last_updated_date	datetime	8	0

### Details\_Experience:

Id	int	4	0
Personnel_No	int	4	0
Pers_no	int	4	1
Company_Name	varchar	500	0
Company_Location	varchar	500	1
Employer_Industry	varchar	500	1
Job_Title	varchar	500	1
Start_Date	varchar	50	1
End_date	varchar	20	1
Exp_Month	float	8	1
exp_field1	varchar	4	1
exp_field2	varchar	4	1
exp_field3	varchar	4	1
exp_field4	varchar	4	1



## **Conclusion:**

The "HR EMPLOYEE INFORMATION" has been successfully completed. The goal of the system is achieved and problems are solved. The package is developed in a manner that it is user friendly and required help is provided at different levels.

The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the company's work standards as well as the economical state of the company.

This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which company's standards and capabilities can be scaled to higher dimensions.