Chapter 1

Introduction to the Organization.

**COMPANY PROFILE**

**Tech Hive** is a Multi Domain Organization, covering almost all major trends of modern day technology. From the day of their establishment, they have been constantly widening their horizons and spreading out their feathers to meet the ever increasing demands of their clients. They deal basically internationally and nationally according to demand of their clients.   
They feel immense honor to introduce themselves as one of the leading- Embedded, Industrial Automation, Smart Software based Application and Product Development Company and mostly deal in LIVE POJECTS in the north Indian hemisphere. They are specialized in **6 MONTHS and 6 WEEKS INDUSTRIAL TRAINING for** Engg. & MCA students. They also provide 100% placement assistance to their students. They are concerned with **Electronics** ( Embedded system , VLSI , PCB Designing ) and for **IT** ( c/c++ .net , JAVA , Oracle , web designing , PHP, MySQL, SQL Server ) and for **Networking** ( MCSE / MCSA , CCNA / CCNP , REDHAT Linux RHCT/RHCE)and many more courses etc. They provide the best quality training with latest technologies and equipments.

They are committed to convert their customer’s ideas into real working applications with the latest technology solutions, in least duration at Tech Hive, over the year; they have developed a core competency to maximize the quality & innovation parameter.

**Objectives of the Company**

* Establish itself as an institute of excellence for imparting education and training to generate quality manpower.
* Facilitate education and training institutes in the non-formal sector.
* Develop a mechanism for dynamic revision of course curricula and development of the learning materials in the textbook, CD-ROM and web based form.
* Impart continuing education/refresher training and corporate training to engineering graduates, working professionals and others.
* Develop and implement new schemes of courses in emerging areas as required by industries and others.
* Undertake develop projects and provide services in IT and related.

Chapter 2

Introduction to Project and Project work

**Project Title:** Job Portal.

**Project Leader:** Mr. Rakesh Sharma.

**Project Duration:** Six Months.

**Project category:** Web Based Application.

Chapter 2.1

Introduction

The project named **Job Portal** is developed by using ASP.NET. This project has been developed in partial fulfillment of Requirements for the degree of B.Tech Punjab Technical University.

Online job portal is a web application built in Asp.net. It provides the candidates ability to register to this application and search jobs, manage their accounts. Each candidate will have their own account with their home page.

On the other hand, companies that are willing to publish the jobs post for candidates, can register to the job portal and get their own account created and can post jobs to portal’s database.

Registered companies can search for the candidates according to their experience, salary required etc. and candidates can also search for the jobs based on multiple criteria. Main aim of this web application is to make a job portal, which is common for job seeker and recruiter.

Moreover, we also look ahead to add more features to this project.

**Statement about the Problem-existing system**

The existing system work manually. The existing system has lot of intricacies within itself and need lot of human effort and paper works. All above the data need to be maintained on ledgers and maintaining this is a tedious and risky process. As the transaction increases, so the data too increases. Therefore, the task of maintaining them increases exponentially. To view a data may need a lot of paper to search.

Some of the negative aspects of the existing system as follow:

**Time consuming:**

Course of action is time consuming. Whenever a need for search arises, the process evolves search through paper records.

**Readability problem:**

Readability of record is constrained. All the records may not be handled or written by the same person. Therefore, the format and style of the records differ and hence it is difficult to understand.

**Reliability:**

Paper records are easily damaged in course of time. The time of paper record is unreliable less that it easily is spoiled.

**Man Power:**

Expenditure is high. Manual system needs added manpower.

**Security:**

Prone to corruption by unauthorized users. Securing of manual system is not fully guaranteed. Inaccuracy can be expected. Data can get easily scratched.

**Objectives of Project**

After thoroughly analyzing the existing system the following objectives have been set:

* Providing user friendly interface
* Keeping project records online
* To get all the information about the topics online
* Easy access of data
* Easy maintenance
* Maintaining data consistency
* Providing better performance
* Increasing the efficiency through automation
* Adequate validation checks for data entry
* Facility to update the data time to time
* Adequate security of the database

## Project Specifications

## Scope:

## It facilitates in two modes one is for employer for recruitment purpose and other is for jobseekers for job placement.

# Users:

## This site can be used by 3 types of users:

# Employer module provides functionalities related to employers. Employers can post vacancies and update the details whenever necessary. Employers can search through applicant resumes based on different criteria.

# Admin module provides administrator related functionalities. Administrator manages entire application and maintains the profiles of applicants and employers.

# User module provides functionalities related to jobseekers. Seekers can post their resumes. They can also update the resume as frequently as required. The applicant can browse through the present vacancies available. Job seekers can also get mail alerts when their resumes are selected by employers.

# Requirements:

# Functional Requirements:

# If the user is an admin, he has to login.

# Admin can delete records related to various employers and seekers.

# The user can upload and reply queries.

# The concept of Master Pages and Content Pages is used.

# Non-functional Requirements:

## Portability:

## The system will be designed to be portable across popular Windows OS.

## Extensibility:

## The system should be extensible to add further information and users for more expansion.

## Re-Usability:

## The system’s code could be reused to add further new features if need to be added in future.

## Reliability and Availability:

## System shall be able to deliver the required in reliable manner.

## Software Upgradeability:

## System is to be developed in phases, so it shall be easily upgradeable to include the new items in the database.

## User Interface Requirements:

**5.1 Log in screen:**

Admin has to first log in to the site.

**5.2 Home page:**

Homepage allows user to browse the entire portal**.**

**5.3 Panels:**

Website is divided into 3 panels for Home page of different users:

* Admin panel
* Employer panel
* Job Seeker panel

**5.3.1** **Admin panel:**

Admin module provides administrator related functionalities. Administrator manages entire application and maintains the profiles of applicants and employers.

**5.3.2 Employer panel:**

Employer module provides functionalities related to employers. Employers can post vacancy details and update the details as and when necessary. Employers can search through applicant resumes based on different criteria.

**5.3.3 Seeker panel:**

This module provides functionalities for job seekers. Applicants can post their resumes with personal and professional details. They can also update the resume as frequently as required. The applicant can browse through the present vacancies available. Job seekers can also get mail alerts when their resumes are selected by employers.

Chapter 2.2

WORKING ENVIRONMENT

**The .Net Frame Work**

**DEFINITION:**

The .Net framework can be defined as a language neutral platform designed to provide a number of rich and powerful application development tools and technologies. It has an extensive class library that provides wide-ranging support for data access.

.Net framework has an independent code execution and management environment called the **Common Language Runtime (CLR)** which ensures that code is safe to run, and provides an abstract layer on top of the operating system, which allows the elements of the .Net framework to run on many operating systems and devices.

**THE .NET VISION:**

The basic idea behind the development of the .Net framework is that a global broadband network will someday globally connect all devices and the software will become a service provided over this network.

The .Net framework provides the foundation and the plumbing on which the Microsoft .Net vision is built. The vision is about having a global network and open standards like XML to make it easier for parties to exchange data and work together. Now devices will use common languages like XML over standardized or shared protocols such as HTTP and these devices will be running a multitude of software on various operating systems. This is not limited to Microsoft but also to other operating systems such as sun, IBM.

The .Net framework provides the tools and the technologies needed to write applications that can seamlessly and easily communicate over the Internet using open standards like XML and SOAP. It also aims at solving the various problems that are faced by developers while working on applications made using the windows DNA. It provides a solution to problems such as registering of components and shutting down applications in order to upgrade them.

**BENEFITS OF USING .NET:**

The windows platform is inherently complex and may pose several problems for programmers. However the .Net platform is used as it provides solutions to many such problems and offers an Internet centric platform.

The following are the benefits of using .Net platform:

1. **Multiple platform Support**

.Net platform has been designed keeping multiple platform support as a key feature. For version 2.0 this means that the code written using the .Net platform can run on all versions of Windows. i.e. Windows 98,95,NT,2000 and so on.

Microsoft has included rich support for all the platforms. Also this code shall also work on any 64-bit processor that may be used by Microsoft later.

It is also expected that .Net shall run on other platforms like UNIX also, however it is not for sure that all functionality will be available for the same.

1. . **Net as a clean start**

On the windows platform, the technologies available often depend on the programming language that they are written in, so they are very different. Depending on the chosen programming language, these technologies can be restrictive.

The problem with this approach is that it makes the Windows platform harder to understand.

Using .Net there is just one simple object oriented way of accessing the functionality of the .Net Framework.

**. Net Framework**

#### ASP

#### MFC

#### VB FORM

#### Windows API

(The best and most commonly used technologies being merged in .Net)

1. **Performance**

. Net has been designed to provide great performance and scalability. For .Net to succeed, the companies must be able to migrate their applications and not suffer due to the way the CLR works. To ensure this CLR converts the code into native machine code. So the conversion can take step-by-step method and will make use of the given machines resources and processor features.

As newer versions of the CLR are released and technologies like windows forms are released, each release will have a better performance and smaller memory footprints. Also .Net has succeeded in replacing completely the older technologies like COM with better and efficient design equivalents. At the heart of the .Net platform is the CLR.

**Microsoft SQL Server 2008:**

Microsoft SQL Server 2008 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online.

**Components of Microsoft SQL Server 2008**

* **Enterprise Manager:** is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.
* **Query Analyzer:** offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.
* **SQL Profiler:** provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.
* **Service Manager:** is used to control the MS-SQL Server (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQL Server Agent processes. An icon for this service normally resides in the system tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.
* **Data Transformation Services (DTS):** provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.
* **Books Online:** is an often overlooked resource provided with SQL Server that contains answers to a variety of administrative, development and installation issues. It's a great resource to consult before turning to the Internet or technical support.

**Features of Microsoft SQL Server 2008**

* **User-Defined Functions:** SQL Server 2000 introduces the long-awaited support for user-defined functions. User-defined functions can take zero or more input parameters and return a single value—either a scalar value like the system-defined functions, or a table result. Table-valued functions can be used anywhere table or view expressions can be used in queries, and they can perform more complex logic than is allowed in a view.
* **Indexed Views:** Views are often used to simplify complex queries, and they can contain joins and aggregate functions. In SQL Server 2000 Enterprise or Developer Edition, you can define indexes on views to improve query performance against the view. When creating an index on a view, the result set of the view is stored and indexed in the database. Existing applications can take advantage of the performance improvements without needing to be modified.

Chapter 2.3

System Requirements

In this step, we analyzed the processing capabilities of the system on which the proposed system would be developed and the specification is divided into two categories namely:

1. **Hardware Specifications**

**Processor**

Minimum: 600 megahertz (MHz) Pentium processor

Recommended: 1 gig hertz (GHz) Pentium processor

**RAM**

Minimum: 192 megabytes (MB)

Recommended: 256 MB

**Hard Disk**

2 GB of available space required on installation drive

1 GB of available space required on system drive

**Display**

Minimum: 800 x 600 256 colors

Recommended: 1024 x 768 High Color - 16-bit

# Technologies to be used

* ASP.NET 3.5 is being used for developing the web pages.
* C#. NET is being used for logic development for the project.
* SQL SERVER is being used for managing the database at server side.

1. **OS required** Microsoft Windows XP / higher version of Windows OS required.

CHAPTER 2.4

Project Design

System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. The plan of the project provides a review of the different modules in which the project has been divided. The modules are designed and tested individually and then merged together to form an integrated project. Three-tier architecture has been adopted in the development of the project.

**Three tier Architecture**

Three-tier is a client-server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

The three-tier model is considered to be software architecture and a software design pattern.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

**Data Tier**

**Definition:** This tier consists of Database Servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

The Data-Tier of our project is completely designed in SQL-Server 2008. It consists of following parts

1. Database Tables

**Database Tables**

Following Database tables have been used in our project.

1. **Emp\_acc\_info:** This table stores the information regarding different employers.
2. **Emp\_company\_info:** This table stores the information regarding different companies.
3. **Emp\_contact\_info:** It contains the information of contacts of individual members.
4. **Emp\_post\_jobs:** It stores all the jobs posted by the employer.
5. **User\_login:** This table stores the information regarding the login of different users.
6. **Contact\_info:** It contains the information of contacts individual members.
7. **Useremp\_detail:** It stores the information of employment details of users.
8. **Userbackground\_edu:** It stores the information of background education of users.
9. **Desired\_job:** It stores the information of regarding desired jobs of users.
10. **Resume:** It stores the information of resumes submitted by users.

**Presentation-Tier**

This is the topmost level of the application. The presentation tier displays information related to such services as browsing job seekers or employer and their profile contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

The main purpose of the presentation layer is to provide the end user an interactive Graphical User Interface.

The main function of the presentation layer is to provide parameters to the application layer and provide proper data formatting to the data returned by application layers. In the presentation layer object of classes formed in application layers are created and appropriate function calls are made to give the project better look.

The concept of master pages and content placeholders has been used in this layer. In addition to it all the validation checks and data formatting is also achieved through this layer.

Chapter 2.5

Data Flow Diagrams

**insmsg**

**Sender**

Message **tblname**

**tblname**

**dispmsg**

**Receiver**

Message

**Private Message**

Member Details **tblname**

**Registered User**

**insmsg**

Formatted Query

Data Query Details **tblname**

**tblname**

Unformatted Query

Data

**Posting a Query**

Category Details **tblname**

**Registered User**

**insmsg**

Formatted Post Thread Details **tblname**

**tblname**

Formulated SQL Query

**Posting a Reply**

User ID Availability

**insmsg**

**New User**

Registration Form registration

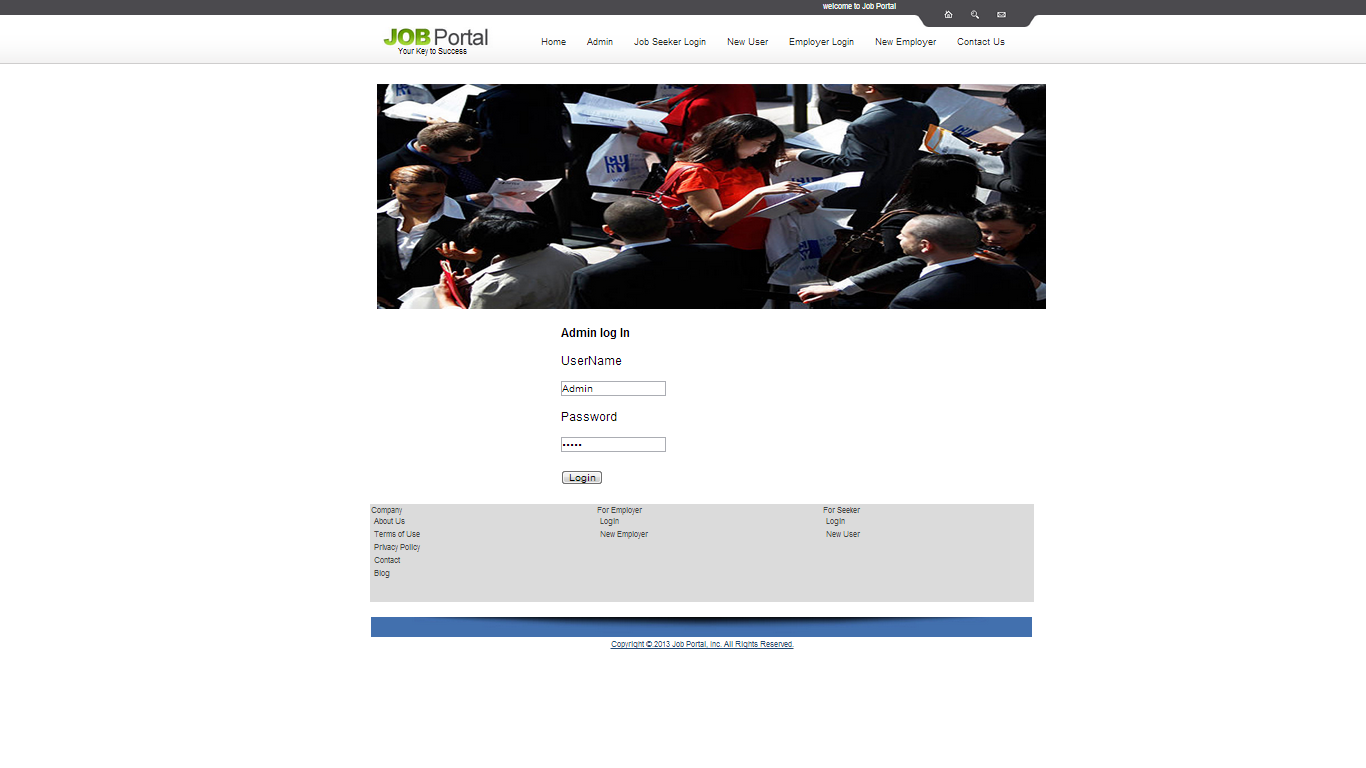
User Detail

**Registration**

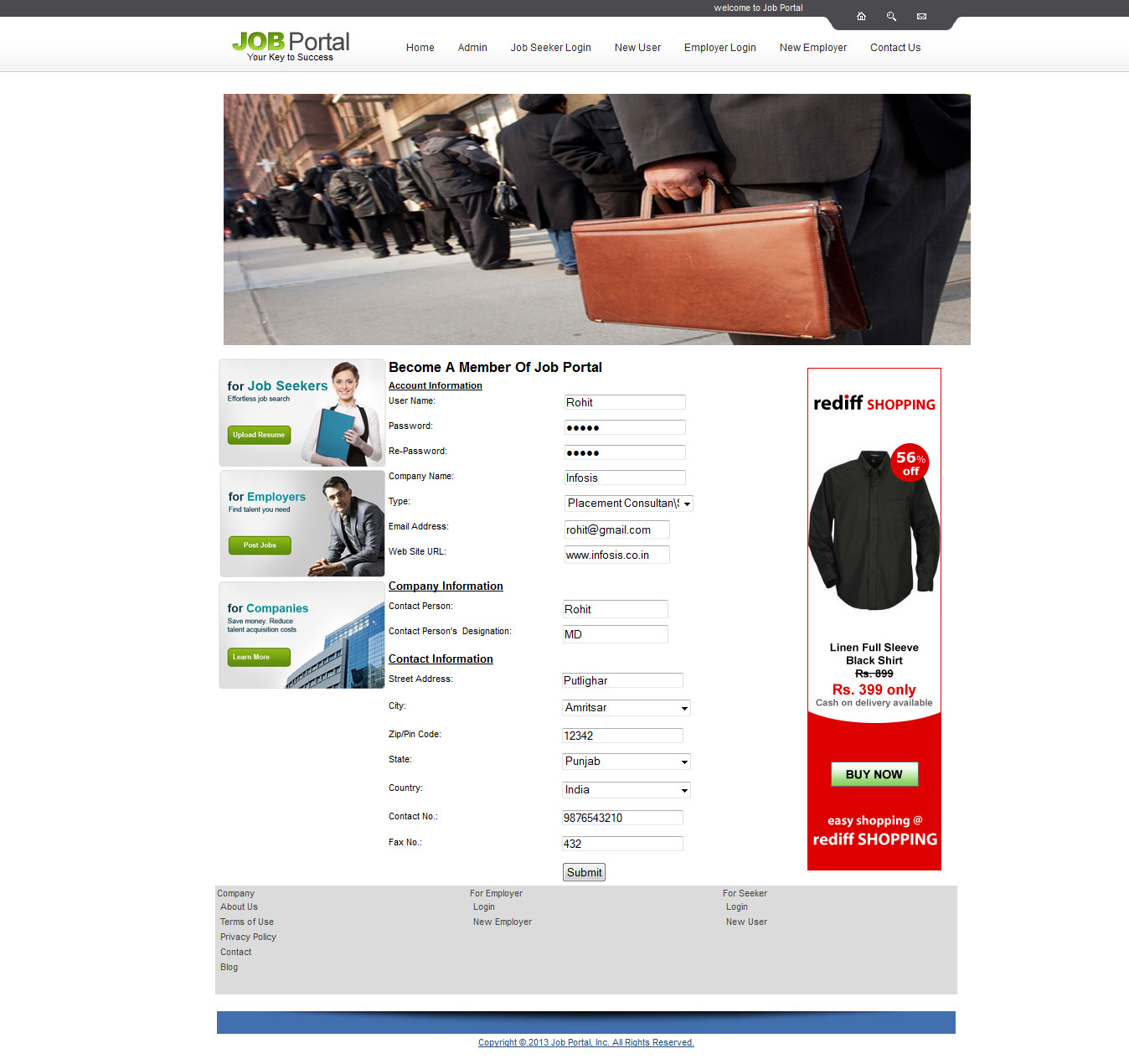
Chapter 3

DESIGN

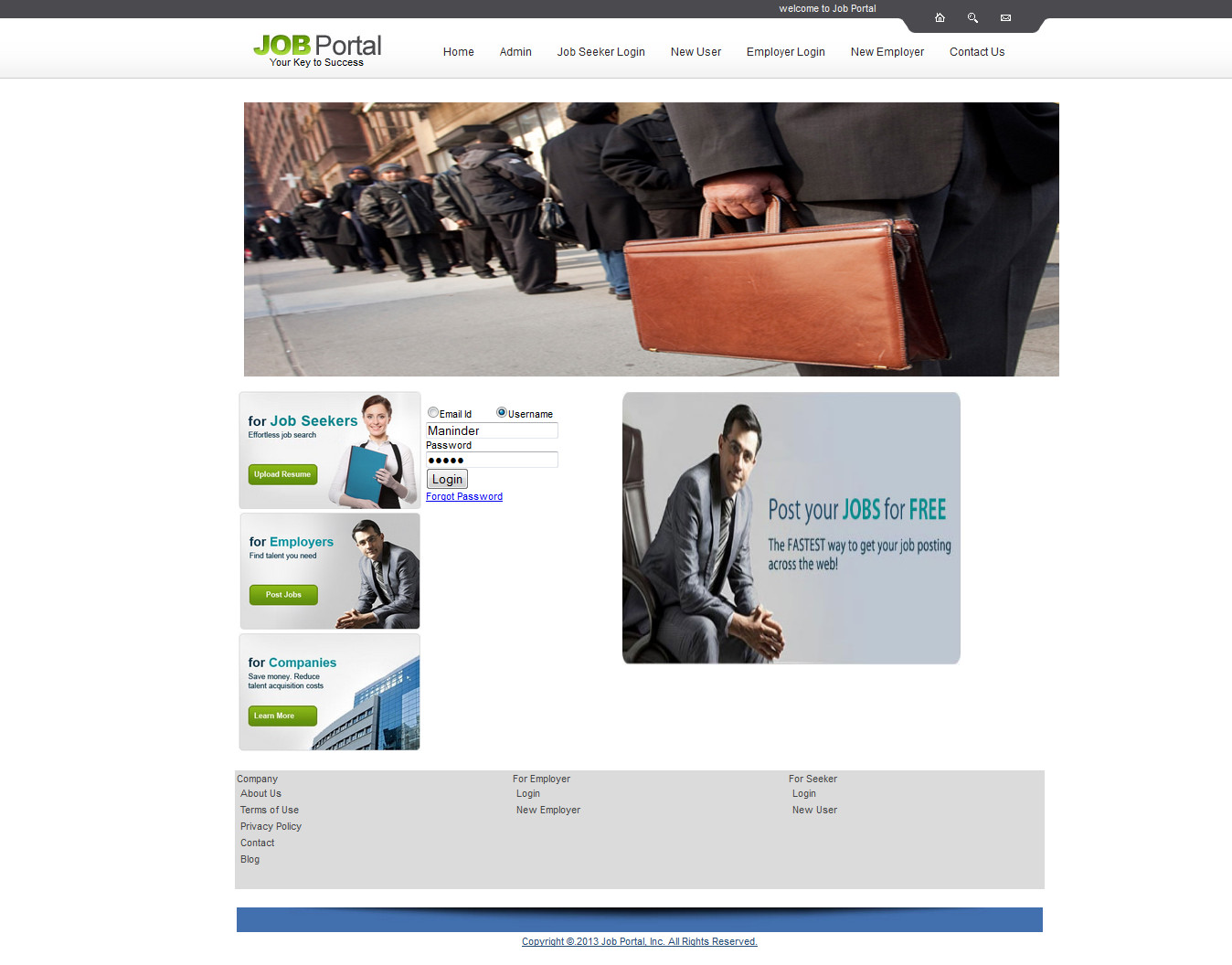
**ADMIN:**



**EMPLOYER REGISTRATION**:



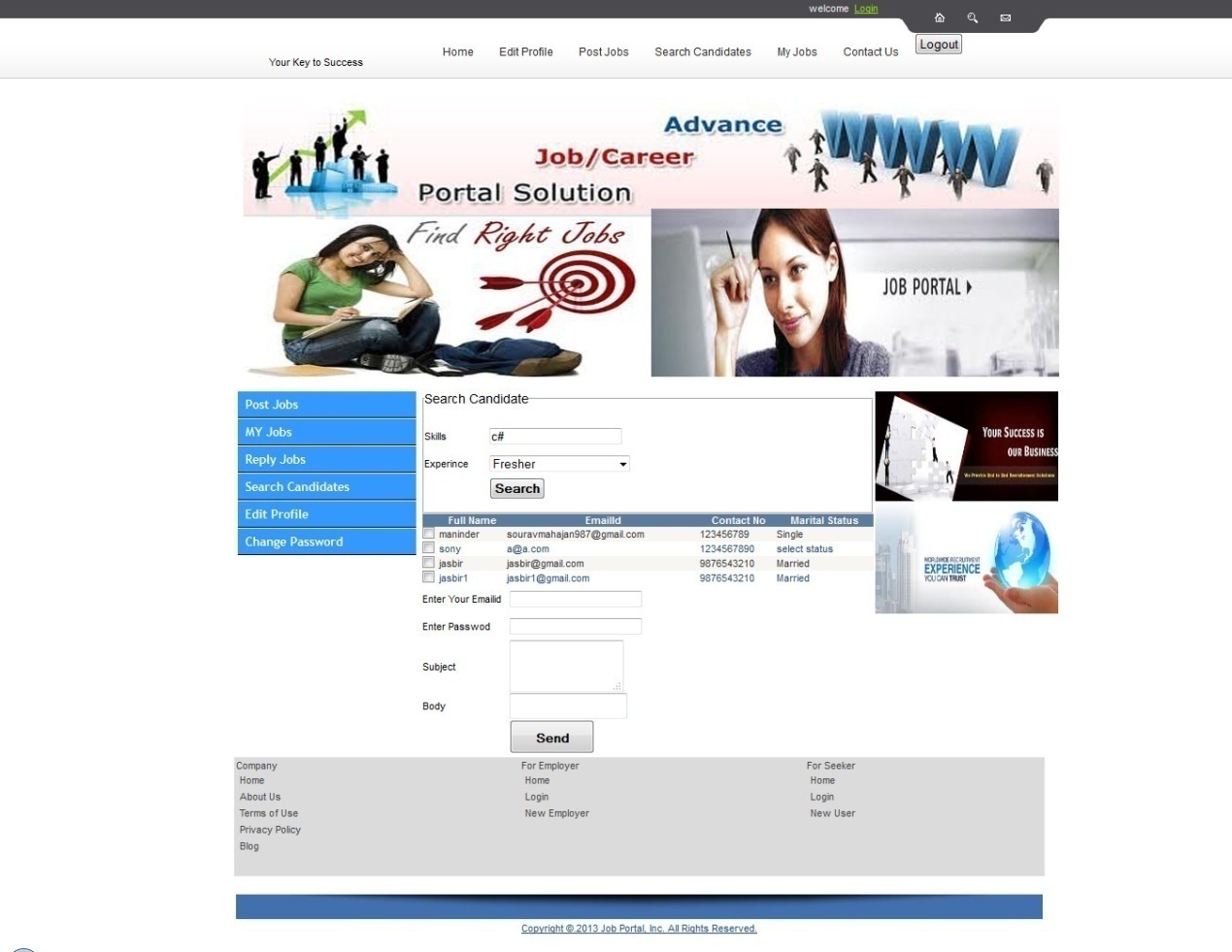
**EMPLOYER LOGIN:**



**POST JOB:**



**SEARCH CANDIDATE**:



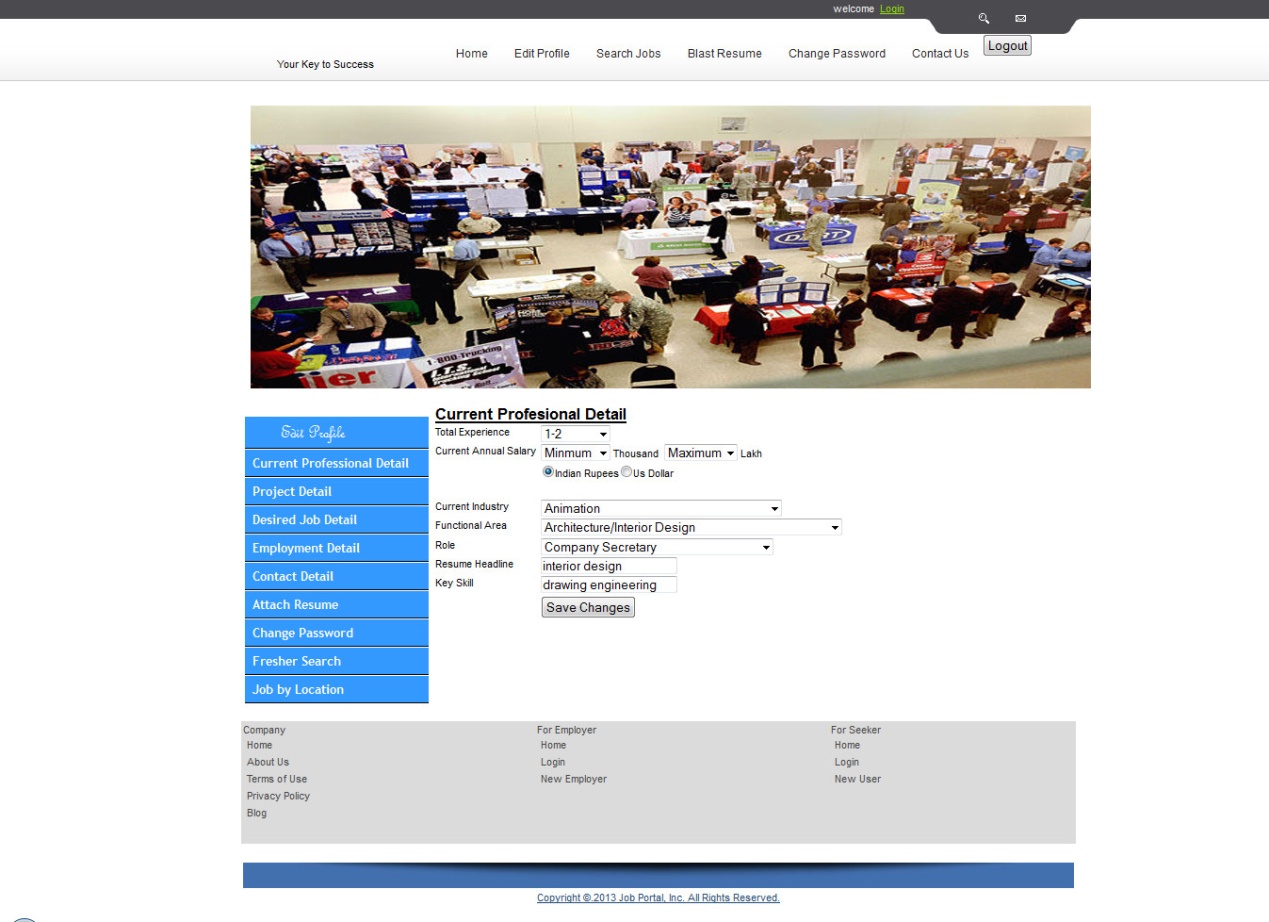
**SEEKER REGISTRATION:**



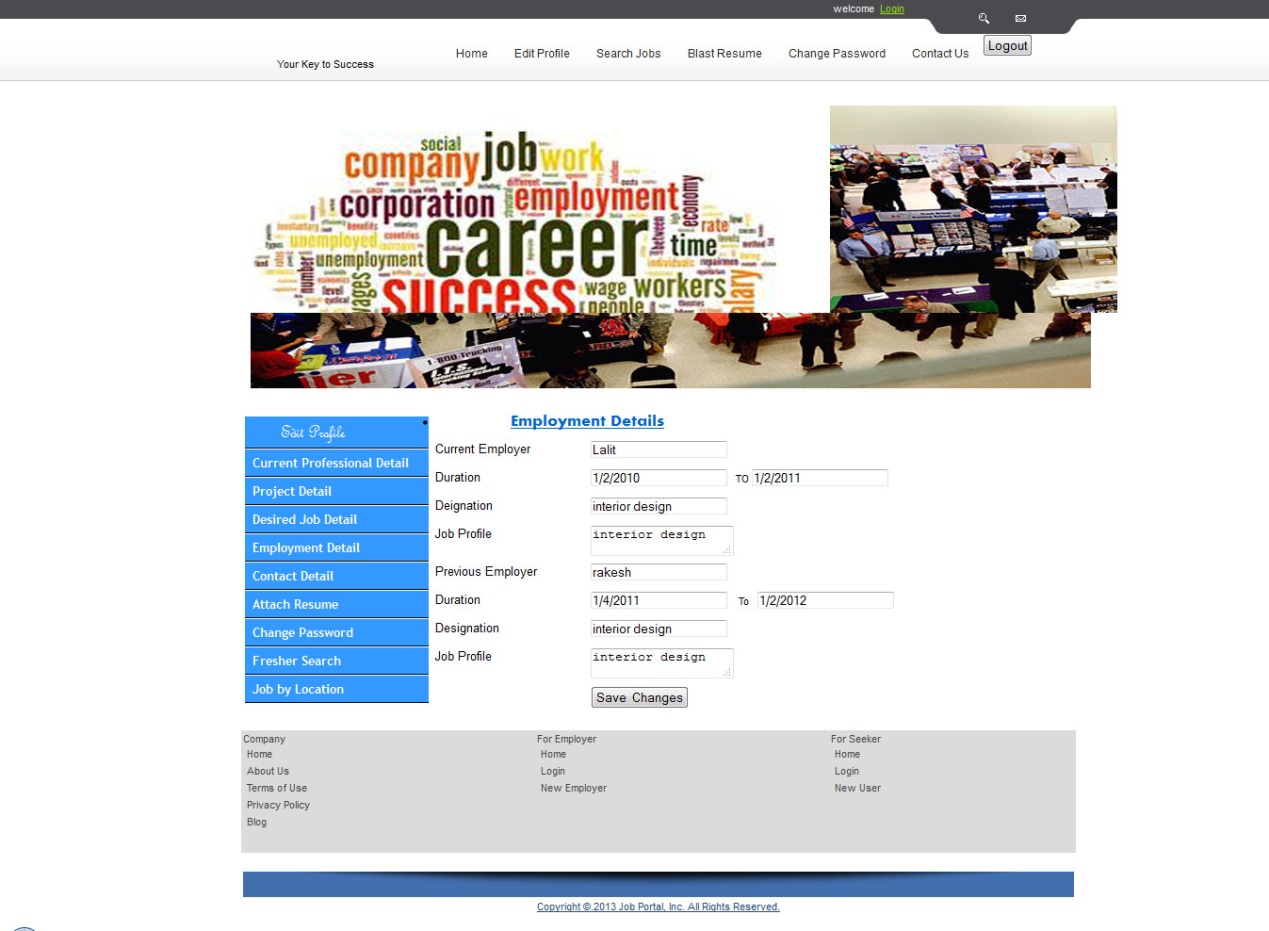
**SEEKER LOGIN:**



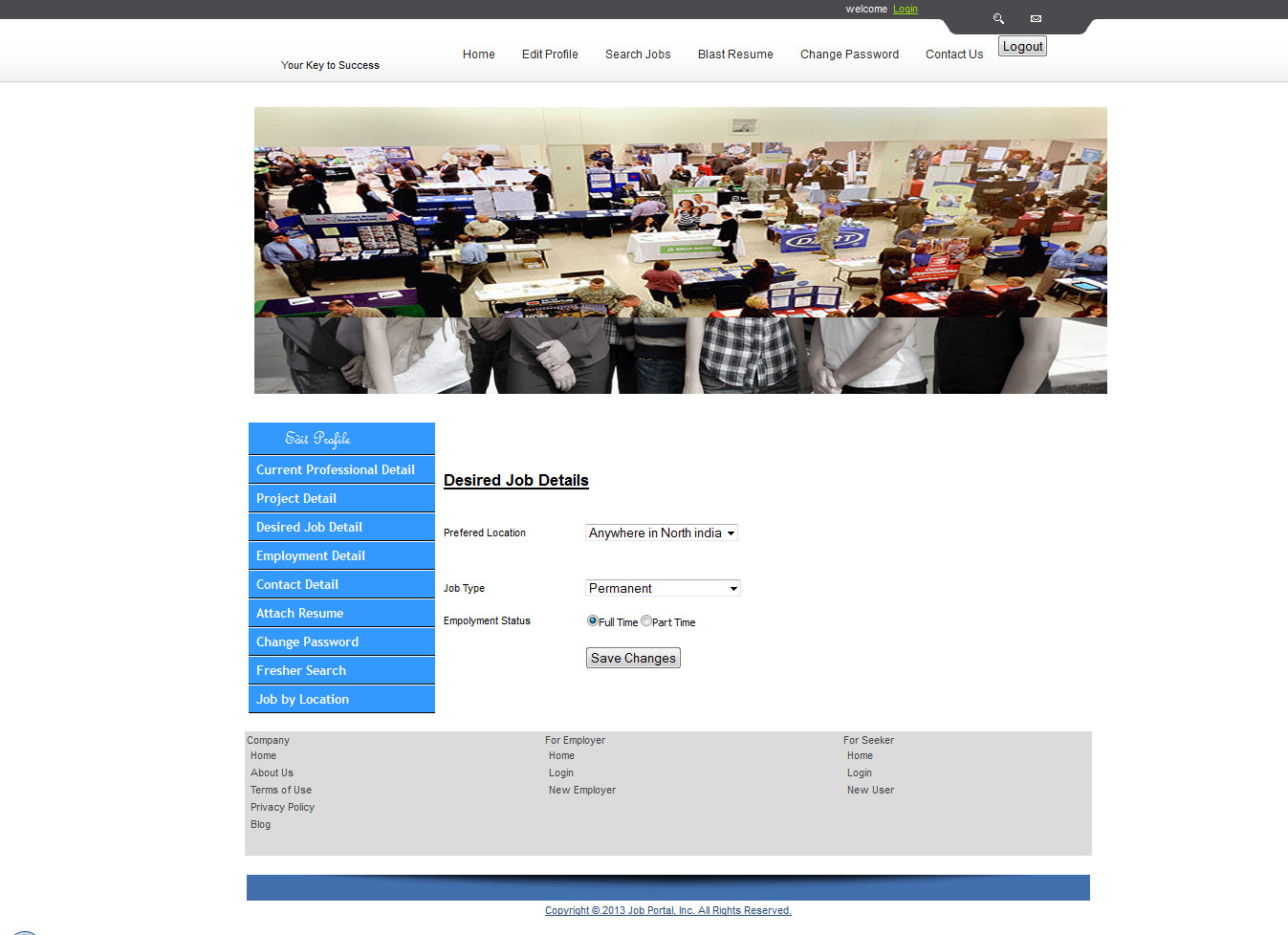
**CURRENT PROFFESIONAL DETAIL:**



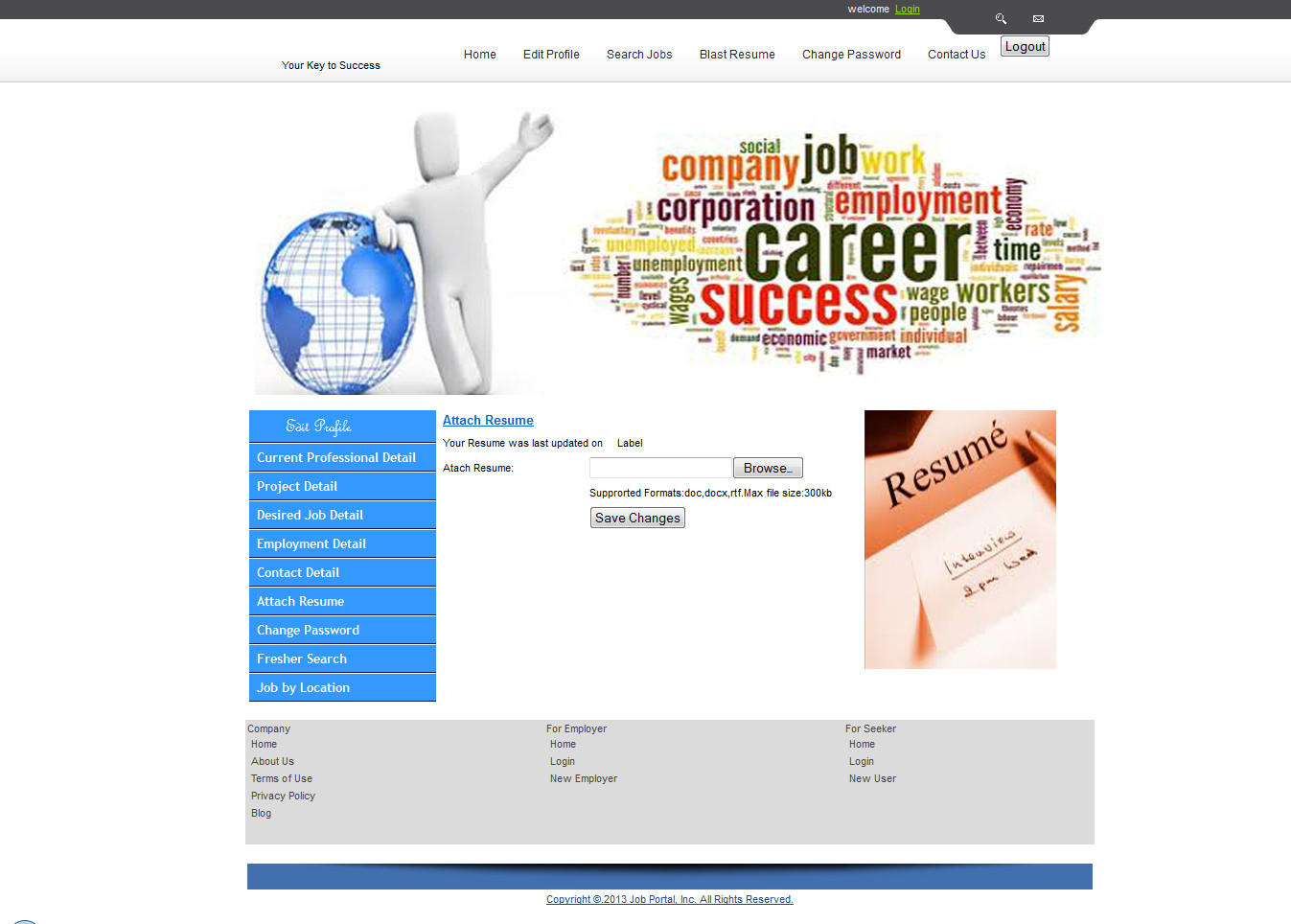
**EMPLOYMENT DETAIL**:



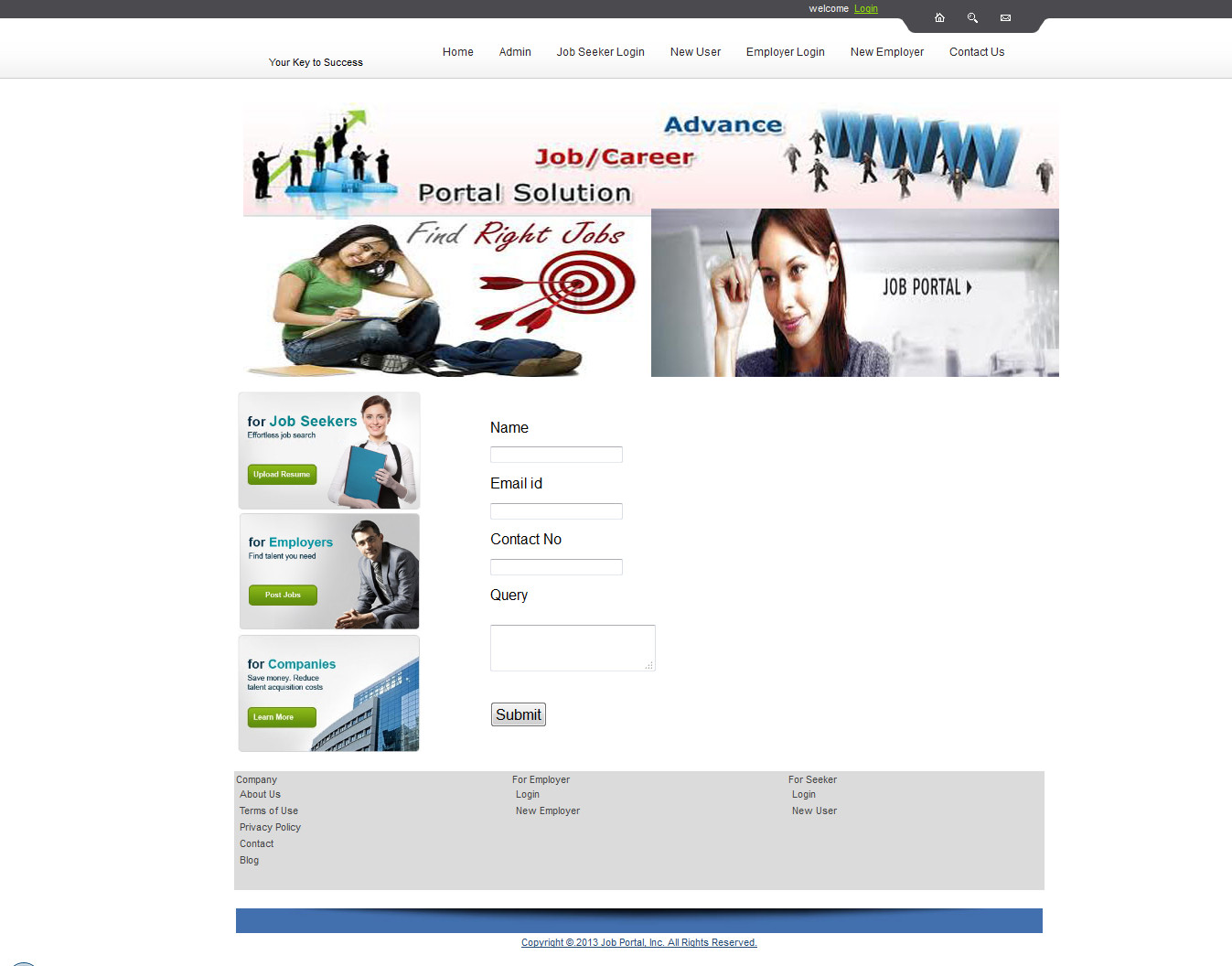
**DESIRED JOB DETAIL:**



**ATTACH RESUME**:



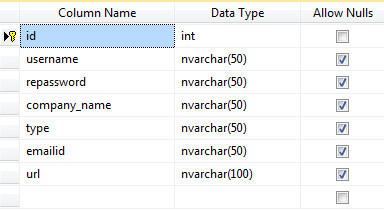
**CONTACT US:**



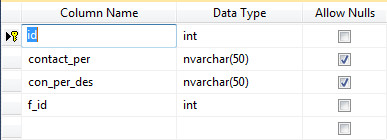
Chapter 4

TABLE

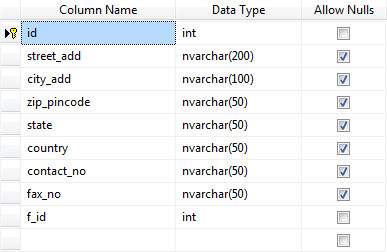
**EMPLOYER ACCOUNT INFO:**



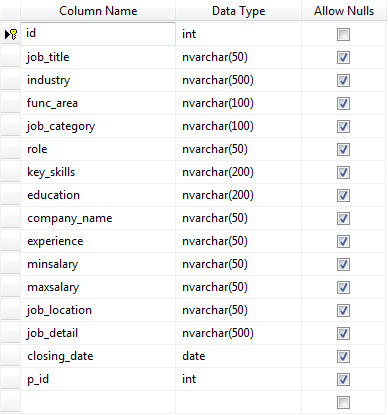
**EMPLOYER COMPANY:**



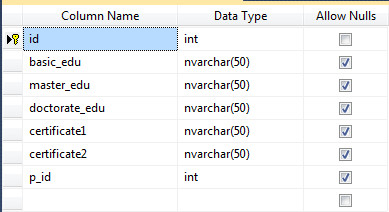
**EMPLOYER CONTACT INFO:**



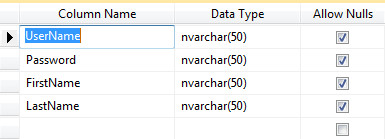
**EMPLOYER POST JOB:**

****

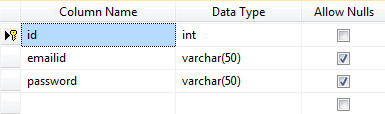
**USER BACKGROUND:**



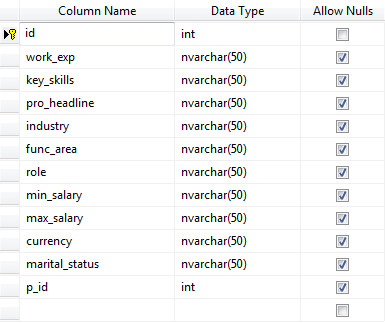
**USER DETAIL:**



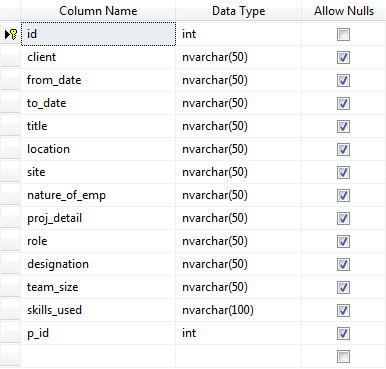
**USER LOGIN:**



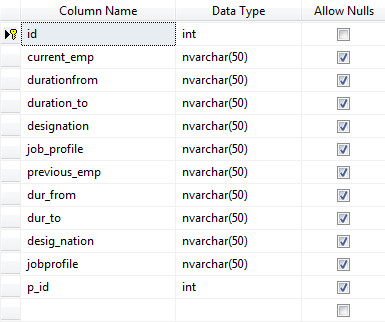
**USER EMP DETAIL:**



**USER PROJECT:**



**AFTER REGISTRATION:**



Chapter 5

CODING

**admin login**

Using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

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

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

if ((TextBox1.Text == "admin") && (TextBox2.Text == "admin"))

{

Response.Redirect("~/admin/home.aspx");

}

else

{

Label4.Text = "Password doesn't exist";

}

}

}

**Employer login**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

public partial class admin\_login : System.Web.UI.Page

{

SqlConnection conn;

SqlCommand cmd;

SqlDataReader dr;

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (RadioButtonList1.SelectedIndex == 1)

{

conn.Open();

cmd = new SqlCommand("select username from emp\_acc\_info where username=@username ", conn);

cmd.Parameters.AddWithValue("@username", TextBox1.Text);

dr = cmd.ExecuteReader();

if (dr.Read())

{

{

if (TextBox1.Text == dr[0].ToString())

dr.Dispose();

cmd = new SqlCommand("select \* from emp\_acc\_info where username=@username AND repassword=@password ", conn);

cmd.Parameters.AddWithValue("@username", TextBox1.Text);

cmd.Parameters.AddWithValue("@password", TextBox2.Text);

dr = cmd.ExecuteReader();

}

if (dr.Read())

{

if (TextBox2.Text == dr[2].ToString())

{

Session["username"] = dr[0].ToString();

Session["welcome"] = dr[1].ToString();

Response.Redirect("~/employer/Default.aspx");

Label2.Text = "login successful";

}

}

else

{

Label2.Text = "Invalid Password";

}

}

else

{

Label2.Text = "Invalid Username ";

}

}

else if (RadioButtonList1.SelectedIndex == 0)

{

conn.Open();

cmd = new SqlCommand("select \* from emp\_acc\_info where emailid=@emailid ", conn);

cmd.Parameters.AddWithValue("@emailid", TextBox1.Text);

dr = cmd.ExecuteReader();

if (dr.Read())

{

dr.Dispose();

cmd = new SqlCommand("select \* from emp\_acc\_info where emailid=@emailid AND repassword=@password ", conn);

cmd.Parameters.AddWithValue("@emailid", TextBox1.Text);

cmd.Parameters.AddWithValue("@password", TextBox2.Text);

dr = cmd.ExecuteReader();

if (dr.Read())

{

Session["username"] = dr[0].ToString();

Session["welcome"] = dr[1].ToString();

Response.Redirect("~/employer/Default.aspx");

}

else

{

Label2.Text = "Invalid Password";

}

}

else

{

Label2.Text = "Invalid Email Id ";

}

}

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

Response.Redirect("~/employer/pwd\_send\_to\_mail.aspx");

}

}

**employee registration**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

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

{

SqlConnection conn;

SqlCommand cmd;

SqlDataReader dr;

int i;

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

conn.Open();

cmd = new SqlCommand("select max(id) from emp\_acc\_info", conn);

dr = cmd.ExecuteReader();

if (dr.Read())

{

i = 1 + Convert.ToInt32(dr[0]);

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

dr.Dispose();

cmd = new SqlCommand("insert into emp\_acc\_info values (@username,@repassword,@company\_name,@type,@emailid,@url)", conn);

cmd.Parameters.AddWithValue("@username", TextBox1.Text);

cmd.Parameters.AddWithValue("@repassword", TextBox3.Text);

cmd.Parameters.AddWithValue("@company\_name", TextBox4.Text);

cmd.Parameters.AddWithValue("@type", DropDownList1.SelectedValue);

cmd.Parameters.AddWithValue("@emailid", TextBox13.Text);

cmd.Parameters.AddWithValue("@url", TextBox6.Text);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into emp\_company\_info values (@contact\_per,@contact\_per\_des,@f\_id)", conn);

cmd.Parameters.AddWithValue("@contact\_per", TextBox7.Text);

cmd.Parameters.AddWithValue("@contact\_per\_des", TextBox8.Text);

cmd.Parameters.AddWithValue("@f\_id",i );

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into emp\_contact\_info values (@street\_add,@city\_add,@zip\_pincode,@state,@country,@contact\_no,@fax\_no,@f\_id)", conn);

cmd.Parameters.AddWithValue("@street\_add", TextBox9.Text);

cmd.Parameters.AddWithValue("@city\_add", DropDownList2.SelectedValue);

cmd.Parameters.AddWithValue("@zip\_pincode", TextBox10.Text);

cmd.Parameters.AddWithValue("@state", DropDownList3.SelectedValue);

cmd.Parameters.AddWithValue("@country", DropDownList4.SelectedValue);

cmd.Parameters.AddWithValue("@contact\_no", TextBox11.Text);

cmd.Parameters.AddWithValue("@fax\_no", TextBox12.Text);

cmd.Parameters.AddWithValue("@f\_id", i);

cmd.ExecuteNonQuery();

Session["welcome"] = TextBox1.Text;

Response.Write("Congratulations " + TextBox1.Text);

TextBox1.Text = string.Empty;

TextBox2.Text = string.Empty;

TextBox3.Text = string.Empty;

TextBox4.Text = string.Empty;

TextBox13.Text = string.Empty;

TextBox6.Text = string.Empty;

TextBox7.Text = string.Empty;

TextBox8.Text = string.Empty;

TextBox9.Text = string.Empty;

TextBox10.Text = string.Empty;

TextBox11.Text = string.Empty;

TextBox12.Text = string.Empty;

DropDownList1.SelectedIndex = 0;

DropDownList2.SelectedIndex = 0;

DropDownList3.SelectedIndex = 0;

DropDownList4.SelectedIndex = 0;

Session["username"] = i;

Response.Redirect("~/employer/Default.aspx");

}

}

**post jobs**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

public partial class employer\_postjobs : System.Web.UI.Page

{

SqlConnection conn;

SqlCommand cmd;

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

if (Session["username"] == null)

{

Response.Redirect("../Default.aspx");

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

conn.Open();

cmd = new SqlCommand("insert into emp\_post\_jobs(job\_title,industry,func\_area,job\_category,role,key\_skills,education,company\_name,experience,minsalary,maxsalary,job\_location,job\_detail,closing\_date,p\_id) values('"+TextBox2.Text+"','"+DropDownList1.SelectedValue+"','"+DropDownList2.SelectedValue+"','"+DropDownList3.SelectedValue+"','"+TextBox3.Text+"','"+TextBox4.Text+"','"+TextBox5.Text+"','"+TextBox11.Text+"','"+DropDownList6.SelectedValue+"','"+DropDownList4.SelectedValue+"','"+DropDownList5.SelectedValue+"','"+DropDownList7.SelectedValue+"','"+TextBox10.Text+"','"+TextBox9.Text+"','"+Convert.ToInt32( Session["username"])+"')", conn);

cmd.ExecuteNonQuery();

Label1.Text=("Congratulations ");

TextBox2.Text = string.Empty;

TextBox3.Text = string.Empty;

TextBox4.Text = string.Empty;

TextBox5.Text = string.Empty;

TextBox9.Text = string.Empty;

TextBox10.Text = string.Empty;

TextBox11.Text = string.Empty;

DropDownList1.SelectedIndex = 0;

DropDownList2.SelectedIndex = 0;

DropDownList3.SelectedIndex = 0;

DropDownList4.SelectedIndex = 0;

DropDownList5.SelectedIndex = 0;

DropDownList6.SelectedIndex = 0;

DropDownList7.SelectedIndex = 0;

conn.Close();

}

}

**Reply seekers**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Net.Mail;

using System.Net;

using System.Configuration;

public partial class employer\_applydetails : System.Web.UI.Page

{

SqlConnection conn;

SqlDataAdapter ADP;

SqlDataReader dr;

SqlCommand cmd;

DataSet ds = new DataSet();

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

if (Session["username"] == null)

{

Response.Redirect("../Default.aspx");

}

conn.Open();

ADP = new SqlDataAdapter("select \* from apply\_details where jobid='" + Convert.ToString(Session["username"]) + "'", conn);

ADP.Fill(ds);

GridView1.DataSource = ds;

GridView1.DataBind();

}

protected void Button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < GridView1.Rows.Count; i++)

{

GridViewRow row = GridView1.Rows[i];

bool ischecked = ((CheckBox)row.FindControl("CheckBox1")).Checked;

if (ischecked == true)

{

string bf = GridView1.Rows[i].Cells[3].Text;

System.Net.Mail.MailAddress toAddress = new System.Net.Mail.MailAddress(bf);

System.Net.Mail.MailAddress fromAddress = new System.Net.Mail.MailAddress("ersourabhm@gmail.com");

System.Net.Mail.MailMessage mm = new System.Net.Mail.MailMessage(fromAddress, toAddress);

mm.Subject = TextBox5.Text;

mm.Body = TextBox4.Text;

mm.IsBodyHtml = true;

mm.BodyEncoding = System.Text.Encoding.UTF8;

sendMail(mm);

}

}

}

private static string sendMail(System.Net.Mail.MailMessage mm)

{

try

{

string smtpHost = "smtp.gmail.com";

string userName = "ersourabhm@gmail.com";//write your email address

string password = "\*\*\*\*\*\*\*\*\*";//write password

System.Net.Mail.SmtpClient mClient = new System.Net.Mail.SmtpClient();

mClient.Port = 587;

mClient.EnableSsl = true;

mClient.UseDefaultCredentials = false;

mClient.Credentials = new NetworkCredential(userName, password);

mClient.Host = smtpHost;

mClient.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;

mClient.Send(mm);

}

catch (Exception ex)

{

System.Console.Write(ex.Message);

}

return "Send Sucessfully";

}

protected void GridView1\_RowUpdating(object sender, GridViewUpdateEventArgs e)

{

string username = GridView1.Rows[e.RowIndex].Cells[2].Text;

Response.ContentType = "Application/Octet-Stream";

Response.AppendHeader("Content-Disposition", "attachment;filename =" + username + ".doc");

Response.TransmitFile(Server.MapPath("~/resume/"+username + ".doc"));

Response.End();

}

}

**seeker login**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

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

{

SqlCommand cmd1;

SqlConnection conn;

SqlCommand cmd;

SqlDataReader dr;

SqlDataReader dr1;

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

}

protected void Button1\_Click(object sender, EventArgs e)

{ conn.Open();

cmd=new SqlCommand("select \* from user\_login where emailid=@emailid", conn);

cmd.Parameters.AddWithValue("@emailid",TextBox1.Text);

dr= cmd.ExecuteReader();

if (dr.Read())

{

if (TextBox1.Text == dr[1].ToString())

{

if (TextBox2.Text == dr[2].ToString())

{

Label3.Text = "login sucessful";

TextBox1.Text = string.Empty;

Session["id"] = dr[0].ToString();

dr.Dispose();

cmd1 = new SqlCommand("select user\_login.id,contact\_info.p\_id,contact\_info.username from contact\_info full join user\_login on user\_login.id=contact\_info.p\_id where user\_login.id='"+Session["id"]+"'", conn);

dr1 = cmd1.ExecuteReader();

if (dr1.Read())

{

Session["welcome"] = dr1[2].ToString();

}

dr1.Dispose();

Response.Redirect("~/jobseeker/CompaniesLOGO.aspx");

}

else

{ Label4.Text="password doesn't match"; }

}

else

{ Label4.Text="wrong emailid"; }

}

else

{

Label4.Text="email doesn't exist";

}}}

**seeker registration**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

public partial class admin\_registration : System.Web.UI.Page

{ string sr; int i;

SqlConnection conn;

SqlCommand cmd;

SqlDataAdapter adp;

SqlDataReader dr;

DataSet ds;

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

{

conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

conn.Open();

cmd = new SqlCommand("select max(id) from user\_login", conn);

dr = cmd.ExecuteReader();

if (dr.Read())

{ i = 1 + Convert.ToInt32(dr[0]); } Panel9.Visible = false;

}

protected void Button1\_Click(object sender, EventArgs e)

{ dr.Dispose();

cmd = new SqlCommand("select user\_login.emailid,contact\_info.username from user\_login join contact\_info on user\_login.id=contact\_info.p\_id where user\_login.emailid='" + TextBox1.Text + "'", conn);

dr = cmd.ExecuteReader();

if (dr.Read())

{ Label27.Text = "Email Id Allready Exist";

Label28.Text = "Username Allready Exist";

}

else

{

{ dr.Dispose();

cmd = new SqlCommand("insert into user\_login values (@emailid,@password)", conn);

cmd.Parameters.AddWithValue("@emailid", TextBox1.Text);

cmd.Parameters.AddWithValue("@password", TextBox3.Text);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into contact\_info values(@username,@current\_location,@country,@mobile\_no,@p\_id)", conn);

cmd.Parameters.AddWithValue("@username", TextBox4.Text);

cmd.Parameters.AddWithValue("@current\_location", DropDownList1.Text);

cmd.Parameters.AddWithValue("@country", DropDownList9.SelectedValue);

cmd.Parameters.AddWithValue("@mobile\_no", TextBox5.Text);

cmd.Parameters.AddWithValue("@p\_id", i);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into useremp\_detail values(@work\_exp,@key\_skills,@pro\_headline,@industry,@func\_area,@role,@min\_salary,@max\_salary,@currency,@marital\_status,@p\_id)", conn);

cmd.Parameters.AddWithValue("@work\_exp", DropDownList2.Text);

cmd.Parameters.AddWithValue("@key\_skills", TextBox6.Text);

cmd.Parameters.AddWithValue("@pro\_headline", TextBox7.Text);

cmd.Parameters.AddWithValue("@industry", DropDownList3.Text);

cmd.Parameters.AddWithValue("@func\_area", DropDownList4.Text);

cmd.Parameters.AddWithValue("@role", TextBox13.Text);

cmd.Parameters.AddWithValue("@min\_salary", DropDownList12.Text);

cmd.Parameters.AddWithValue("@max\_salary", DropDownList11.Text);

cmd.Parameters.AddWithValue("@currency", RadioButtonList1.SelectedValue);

cmd.Parameters.AddWithValue("@marital\_status", DropDownList15.Text);

cmd.Parameters.AddWithValue("@p\_id", i);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into userbackground\_edu values(@basic\_edu,@master\_edu,@doctorate\_edu,@certificate1,@certificate2,@p\_id)", conn);

cmd.Parameters.AddWithValue("@basic\_edu", DropDownList8.Text);

cmd.Parameters.AddWithValue("@master\_edu", DropDownList6.Text);

cmd.Parameters.AddWithValue("@doctorate\_edu", DropDownList7.Text);

cmd.Parameters.AddWithValue("@certificate1", TextBox8.Text);

cmd.Parameters.AddWithValue("@certificate2", TextBox11.Text);

cmd.Parameters.AddWithValue("@p\_id", i);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into desired\_job values(@preferred\_loc,@job\_title,@job\_type,@emp\_status,@p\_id)", conn);

cmd.Parameters.AddWithValue("@preferred\_loc", DropDownList13.Text);

cmd.Parameters.AddWithValue("@job\_title", TextBox12.Text);

cmd.Parameters.AddWithValue("@job\_type", DropDownList14.Text);

cmd.Parameters.AddWithValue("@emp\_status", RadioButtonList2.SelectedValue);

cmd.Parameters.AddWithValue("@p\_id", i);

cmd.ExecuteNonQuery();

cmd = new SqlCommand("insert into resume values(@resume,@f\_id)", conn);

cmd.Parameters.AddWithValue("@resume", TextBox4.Text + ".doc");

cmd.Parameters.AddWithValue("@f\_id", i);

cmd.ExecuteNonQuery();

string location = Server.MapPath("~/resume/");

if (FileUpload1.HasFile)

{

FileUpload1.SaveAs(location + TextBox4.Text + ".doc");

}

conn.Close();

Session["id"] = Convert.ToInt32(i);

Session["welcome"] = TextBox4.Text;

Response.Redirect("after\_registration.aspx");

}}}

**search jobs**

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

public partial class advance\_search : System.Web.UI.Page

{ SqlConnection conn; SqlCommand cmd; SqlDataReader dr;

SqlDataAdapter adp; string l;

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

{ conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["key"].ConnectionString;

if (Session["id"] == null)

{ Response.Redirect("../Default.aspx"); }

}

protected void Button1\_Click(object sender, EventArgs e)

{ conn.Open();

adp = new SqlDataAdapter("select \* from emp\_post\_jobs where key\_skills='" + TextBox1.Text + "'and experience='" + DropDownList1.SelectedValue + "'and job\_location='" + DropDownList6.SelectedValue + "'and job\_category='" + DropDownList2.SelectedValue + "' and industry='" + DropDownList5.SelectedValue + "' order by id desc", conn);

adp.SelectCommand.ExecuteNonQuery();

DataSet ds = new DataSet();

adp.Fill(ds);

GridView1.DataSource = ds;

GridView1.DataBind();

}

protected void GridView1\_RowUpdating1(object sender, GridViewUpdateEventArgs e)

{ SqlDataReader dr1; string username;

conn.Open();

SqlDataAdapter adp2 = new SqlDataAdapter("select emailid from user\_login where id='" +Convert.ToInt32( Session["id"]) + "'", conn);

SqlDataReader dr2;

dr2 = adp2.SelectCommand.ExecuteReader();

if (dr2.Read())

{ l = dr2[0].ToString(); } dr2.Dispose();

SqlCommand cmd1 = new SqlCommand("select username from contact\_info where p\_id='" + Convert.ToString(Session["id"]) + "'", conn);

dr1 = cmd1.ExecuteReader();

if (dr1.Read())

{ username = dr1["username"].ToString();

GridViewRow rows = GridView1.Rows[e.RowIndex];

string j = ((Label)rows.FindControl("Label11")).Text;

string g = ((LinkButton)rows.FindControl("LinkButton1")).Text;

SqlCommand cmd2 = new SqlCommand("insert into apply\_details(userid,username, jobid,job\_title,date,emailed) values('" + Convert.ToInt32(Session["id"])+ "', '" +username+"','"+Convert.ToInt32( j)+"','"+g+ "', '"+ DateTime.Now+ "','"+l+ "'”, conn);

dr1.Dispose();

cmd2.ExecuteNonQuery();

conn.Close(); }}}

Chapter 6

CONCLUSION

The system has been developed for the given condition and is found working effectively. The developed system is flexible and changes can be made easily whenever required. Using the facilities and functionalities of .Net, the software has been developed in a neat and simple manner, thereby reducing the operator’s work.

The speed and accuracy are maintained in proper way. The user-friendly nature of this software developed in .Net framework is very easy to work with both the higher management as well as other users with little knowledge of computer. The results obtained were fully satisfactory from the user point of view.

The system has been verified with valid as well as invalid data in each manner. The system is run with an insight into the necessary modifications that may be required in the future. Hence, the system can be maintained successfully.

Chapter 7

Future Scope

The current project has been developed in limited time period of six weeks. In those six weeks, we have learned a new technology as well as implemented it. Therefore, the actual time for the development of the project was very limited. Despite of limited time the project has been developed in such a way that it can be easily expanded. For example, the Data-Tier of this project contains some stored procedures, which are not implemented in the project; these stored procedures are general in nature and can prove their importance by saving hard work during expansion of the project. Similarly, there are many methods in the Application-Tier that are not utilized in the project these methods can also prove handy during the expansion.

Chapter 8

BIBLIOGRAPHY

**CatalogueS**

Training sessions conducted by company itself.

**BOOKS**

Database management system Vipin C. Desai

System Analysis and Design Elias M. Awad

SQL Server Microsoft Press

ASP.NET Wrox publications

Javascript Ivan Byross

**WEBSITES**

[www.dotnetspider.com](http://www.dotnetspider.com)

[www.gotdotnet.com](http://www.gotdotnet.com)

[www.onlinetemplates.org](http://www.onlinetemplates.org)

[www.webopedia.com](http://www.webopedia.com)

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)