**GRIEVIENCE REDRESSAL SYSTEM**

**Submitted in partial fulfillment of the requirements**

**for the award of degree of**

**B.Sc (INFORMATION TECHNOLOGY)**

**of Madurai Kamaraj University**

**Submitted By**

**K.ANU PRIYA (2017IT05)**

**Under The Guidance Of**

**Mrs. V. MAGESHWARI M.Sc., M.C.A., M.Phil.**

# Hod of information technology



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**FATIMA COLLEGE(AUTONOMOUS)**

**MADURAI**

**November 2019**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**BONAFIDE CERTIFICATE**

This is to certify that the Project Report entitled **“GRIEVIENCE REDRESSAL”** is the bonafide record work done by, **K.ANU PRIYA (2017I05)** in partial fulfillment for the award of the degree B.Sc(INFORMATION TECHNOLOGY), Fatima College, Madurai.

Submitted for the viva-voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Guide Head of the Department

**DECLARATION**

I hereby declare that the project work entitled **“GRIVIENCE REDRESSAL”** is a project report of the original work done by myself. This project work is submitted to Fatima College (Affiliated to Madurai Kamaraj University) in partial fulfillment of the degree of Bachelor of Science during the academic year 2015-2018.

I declare that this dissertation or any part thereof has not been submitted for getting any degree or diploma from any university or college.

Place:

Date:

Signature

(K.ANU PRIYA)

**ACKNOWLEDGEMENT**

I am grateful to mention the people whose constant inspiration, guidance and blessings made my project a successful one.

I thank god for his substantial blessings and mercy at all stages of the completion of my project. Taking this opportunity, I thank my parents and friends for their sacrifices in supporting me.

I feel myself honored to place warm salutation to the management of **FATIMA COLLEGE, MADURAI** which gave me the opportunity to have the strong baseincomputer and technical knowledge.

I express my sincere thanks to **REV.DR CELINE SAHAYA MARY., Principal, Fatima College, Madurai** for giving me a comfortable environment**.**

I express my sincere thanks to **sister. Arul mary, Vice Principal** for her extensive support rendered to carry out this project successfully.

I owe my special thanks to **Mrs. V. Mageshwari, M.Sc., M.Phil., HOD, Department of IT,** for her valuable guidance and encouragement.

I owe my special thanks to **Mrs. V.MAGESHWARI, M.Sc., M.C.A., M.Phil.,** for her valuable guidance and encouragement

I like to thank all the staff members in IT Department of Fatima College, for their valuable suggestions at different stages of this project work.

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Title** | **Page No** |
| 1 | **1. INTRODUCTION**  1.1 Abstract | 2 |
| 2 | **2. SYSTEM ANALYSIS**  2.1 Existing System  2.2 Proposed System  2.2.1 Module Description | 3  4  5 |
| 3 | **3. SYSTEM SPECIFICATION**  3.1 Hardware Requirement  3.2 Software Requirement  3.3 Software Description | 7  7  8 |
| 4 | **4. SYSTEM DESIGN**  4.1 Data Flow Diagram  4.2 ER Diagram  4.3 Table Structure | 16  17  18  19 |
| 5 | **5. PROJECT IMPLEMENTATION**  5.1 Sample Coding  5.2 Sample Output | 21  31 |
| 6 | **6. TESTING**  6.1 Unit Testing  6.2 Integration Testing | 37  38 |
| 7 | **7. CONCLUSION**  7.1 Benefits  7.2 Future Enhancement | 39  39 |
| 8 | **8. BIBLIOGRAPHY** | 40 |

1. **INTRODUCTION**
2. **INTRODUCTION**

Hospital are the essential part of our lives which provides us with the best medical facilities for various sickness, it may be due to the change in climatic conditions, stress (emotional trauma) etc. It is necessary for the hospital to keep track of all activities and records day in and day out of its patient, doctors, nurses and other staffs that keeps the hospital in its operation.

Keeping track of all activities and reports on paper is very inefficient and time consuming and also error prone. Keeping records on paper is a traditional base system that sometimes do not make it robust, in any case of damage all files will be lost that will cost a lot to the organization. Day in and day out many people visit the hospital and when using the traditional base system it make it unreliable in the sense that it will take longer time to enter or access data and also maintaining. It is not economically and technically feasible to maintain these records on paper.

Thus keeping the working on the manual system we have develop an automated version of the manual system called “hospital Management System”. Patient Management System is designed specifically to manage episodes of care quickly and safely in demanding.

**1.1ABSTRACT**

The main aim of the project is to provide a paper-less and also providing low-cost reliable of automation of a reliable existing system. The system also provides excellent security of data at every level of user-system interaction and also provides robust and reliable storage and backup facilities.

The project “hospital management system” is aimed to develop to maintain the day –to-day state of admission/discharge of patients, list of patients, reports generation, and etc. It is designed to achieve the following objectives:

1. To computerize all details regarding patient details and hospital details.
2. Scheduling the appointment of patient with doctors to make it convenient for both.
3. The information of the patients should be kept up to date and there record should be kept in the system for historical purposes.

**2. SYSTEM ANALYSIS**

**2.1EXISTING SYSTEM**

The existing system is a manual system. Normally existing all the Process are Conducted and Written Manually. It takes More Time and make manual mistake. A printed copy of the same is maintained. Security plays a major issue. The details and reports would be viewed and modified by any unauthorized user.

**Need for New System**

* User-friendly application, which gives error, messages corresponding to the context.
* To make the data entry and updating more flexible.
* To make information retrieval easy.
* Provide details through user friendly GUIs.
* Avoid data redundancy and data inconsistency.

**2.2PROPOSED SYSTEM**

. It should enable easier and faster report generation as well as quick retrieval of information .In proposed method the details of the patients is computerized. The problems of the patients are noted and they are given the same token number.

**2.2.1MODULE DESCRIPTION**

* Login
* Patient Data Bank
* Doctor/admin Desk
* Patient Information Review
* Administrative Services

**5.4 MODULE DESCRIPTION**

This system having following modules,

**USER**

**Login**

This form collects the user id and password from the user. If the entry is correct then the user will allow entering the site. If the entry found to be incorrect, the user will not allow to viewing the site.

**Patient Data Bank**

The patient data bank module is used to register new patient, keep the current healthcare status and also keep address information also.

**Doctor Desk**

Doctor Desk module is used to doctor maintains patient oriented healthcare information and keep the updated health information also. The process of patient health care information maintained by each every patient for whom come to treatment.

**Patient History**

The patient history module is used to show the patient healthcare information. The each and every patient having unique ID, the registered Identity will be extract his valid information from the healthcare server.

**ADMIN**

**Login**

This form collects the user id and password from the Admin. If the entry is correct then the user will allow entering the site. If the entry found to be incorrect, the user will not allow to viewing the site.

**Doctor Entry**

This is module is used add doctor information and providing unique login identities.

**3.SYSTEM SPECIFICATION**.

3.1HARDWAREREQUIREMENT

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

3.2 SOFTWARE REQUIREMENTS

* Operating system : Windows 7/8.
* Coding Language : ASP.NET C#
* Tool : Visual Studio
* Data Base : SQL Server 2005

3.3SOFTWARE DESCRIPTION

THE .NET FRAMEWORK:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

OBJECTIVES OF .NET FRAMEWORK:

* To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
* To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
* Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications. To make communication on distributed environment to ensure that code be accessed by the .NET Framework can integrate with any other code.

COMPONENTS OF .NET FRAMEWORK:

***THE COMMON LANGUAGE RUNTIME (CLR):***

The common language runtime is the foundation of the .NET Framework. It manages code at execution time, providing important services such as memory management, thread management, and remoting and also ensures more security and robustness. 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 .NET FRAME WORK CLASS LIBRARY:***

It is a comprehensive, object-oriented collection of reusable types used to develop applications ranging from traditional command-line or graphical user interface (GUI) 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.

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 to enables embeds managed components or Windows Forms controls in HTML documents.

***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 are all run on CLR.

* Security.
* Robustness.
* Productivity.
* Performance.

**Security:**

The runtime enforces code access security. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich. With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin to perform file-access operations, registry-access operations, or other sensitive functions.

**ROBUSTNESS:**

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 managed environment of the runtime eliminates many common software issues.

**PRODUCTIVITY:**

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.

**PERFORMANCE:**

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. Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS).

**FEATURES OF ASP.NET**

##### **ASP.NET**

ASP.NET is the next version of Active Server Pages (ASP); it is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax compatible, it also provides a new programming model and infrastructure for more secure, scalable, and stable applications.

ASP.NET is a compiled, NET-based environment, we can author applications in any .NET compatible language, including Visual Basic .NET, C#, and JScript .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can easily access the benefits of these technologies, which include the managed common language runtime environment (CLR), type safety, inheritance, and so on.

ASP.NET has been designed to work seamlessly with WYSIWYG HTML editors and other programming tools, including Microsoft Visual Studio .NET. Not only does this make Web development easier, but it also provides all the benefits that these tools have to offer, including a GUI that developers can use to drop server controls onto a Web page and fully integrated debugging support. Developers can choose from the following two features when creating an ASP.NET application. Web Forms and Web services, or combine these in any way they see fit. Each is supported by the same infrastructure that allows you to use authentication schemes, cache frequently used data, or customize your application's configuration, to name only a few possibilities. Web Forms allows us to build powerful forms-based Web pages. When building these pages, we can use ASP.NET server controls to create common UI elements, and program them for common tasks. These controls allow we to rapidly build a Web Form out of reusable built-in or custom components, simplifying the code of a page.

An XML Web service provides the means to access server functionality remotely. Using Web services, businesses can expose programmatic interfaces to their data or business logic, which in turn can be obtained and manipulated by client and server applications. XML Web services enable the exchange of data in client-server or server-server scenarios, using standards like HTTP and XML messaging to move data across firewalls. XML Web services are not tied to a particular component technology or object-calling convention.

As a result, programs written in any language, using any component model, and running on any operating system can access XML Web services

Each of these models can take full advantage of all ASP.NET features, as well as the power of the .NET Framework and .NET Framework common language runtime. Accessing databases from ASP.NET applications is an often-used technique for displaying data to Web site visitors. ASP.NET makes it easier than ever to access databases for this purpose. It also allows us to manage the database from your code .

ASP.NET provides a simple model that enables Web developers to write logic that runs at the application level. Developers can write this code in the global.aspx text file or in a compiled class deployed as an assembly. This logic can include application-level events, but developers can easily extend this model to suit the needs of their Web application.

ASP.NET provides easy-to-use application and session-state facilities that are familiar to ASP developers and are readily compatible with all other .NET Framework APIs.ASP.NET offers the IHttpHandler and IHttpModule interfaces. Implementing the IHttpHandler interface gives you a means of interacting with the low-level request and response services of the IIS Web server and provides functionality much like ISAPI extensions, but with a simpler programming model. Implementing the IHttpModule interface allows you to include custom events that participate in every request made to your application.

ASP.NET takes advantage of performance enhancements found in the .NET Framework and common language runtime. Additionally, it has been designed to offer significant performance improvements over ASP and other Web development platforms. All ASP.NET code is compiled, rather than interpreted, which allows early binding, strong typing, and just-in-time (JIT) compilation to native code, to name only a few of its benefits. ASP.NET is also easily factorable, meaning that developers can remove modules (a session module, for instance) that are not relevant to the application they are developing.

ASP.NET provides extensive caching services (both built-in services and caching APIs). ASP.NET also ships with performance counters that developers and system administrators can monitor to test new applications and gather metrics on existing applications.

Writing custom debug statements to your Web page can help immensely in troubleshooting your application's code. However, it can cause embarrassment if it is not removed. The problem is that removing the debug statements from your pages when your application is ready to be ported to a production server can require significant effort.

ASP.NET offers the Trace Context class, which allows us to write custom debug statements to our pages as we develop them. They appear only when you have enabled tracing for a page or entire application. Enabling tracing also appends details about a request to the page, or, if you so specify, to a custom trace viewer that is stored in the root directory of your application. The .NET Framework and ASP.NET provide default authorization and authentication schemes for Web applications. we can easily remove, add to, or replace these schemes, depending upon the needs of our application .

ASP.NET configuration settings are stored in XML-based files, which are human readable and writable. Each of our applications can have a distinct configuration file and we can extend the configuration scheme to suit our requirements.

**DATA ACCESS WITH ADO.NET:**

As you develop applications using ADO.NET, you will have different requirements for working with data. You might never need to directly edit an XML file containing data - but it is very useful to understand the data architecture in ADO.NET.

ADO.NET offers several advantages over previous versions of ADO:

* Interoperability
* Maintainability
* Programmability
* Performance Scalability

***INTEROPERABILITY:***

ADO.NET applications can take advantage of the flexibility and broad acceptance of XML. Because XML is the format for transmitting datasets across the network, any component that can read the XML format can process data. The receiving component need not be an ADO.NET component.

The transmitting component can simply transmit the dataset to its destination without regard to how the receiving component is implemented. The destination component might be a Visual Studio application or any other application implemented with any tool whatsoever.

The only requirement is that the receiving component be able to read XML. SO, XML was designed with exactly this kind of interoperability in mind.

***MAINTAINABILITY:***

In the life of a deployed system, modest changes are possible, but substantial, Architectural changes are rarely attempted because they are so difficult. As the performance load on a deployed application server grows, system resources can become scarce and response time or throughput can suffer. Faced with this problem, software architects can choose to divide the server's business-logic processing and user-interface processing onto separate tiers on separate machines. In effect, the application server tier is replaced with two tiers, alleviating the shortage of system resources. If the original application is implemented in ADO.NET using datasets, this transformation is made easier.

ADO.NET data components in Visual Studio encapsulate data access functionality in various ways that help you program more quickly and with fewer mistakes.

***PERFORMANCE:***

ADO.NET datasets offer performance advantages over ADO disconnected record sets. In ADO.NET data-type conversion is not necessary.

***SCALABILITY:***

ADO.NET accommodates scalability by encouraging programmers to conserve limited resources. Any ADO.NET application employs disconnected access to data; it does not retain database locks or active database connections for long durations.

**VISUAL STUDIO .NET:**

Visual Studio .NET is a complete set of development tools for building ASP Web applications, XML Web services, desktop applications, and mobile applications In addition to building high-performing desktop applications, you can use Visual Studio's powerful component-based development tools and other technologies to simplify team-based design, development, and deployment of Enterprise solutions. Visual Basic .NET, Visual C++ .NET, and Visual C# .NET all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions.

In addition, these languages leverage the functionality of the .NET Framework and simplify the development of ASP Web applications and XML Web services.

Visual Studio supports the .NET Framework, which provides a common language runtime and unified programming classes; ASP.NET uses these components to create ASP Web applications and XML Web services. Also it includes MSDN Library, which contains all the documentation for these development tools.

**MS-SQL SERVER 2005:**

SQL Server 2008 was released on August 6, 2008, announced to the SQL Server Special Interest Group at the ESRI 2008 User's Conference on August 6, 2008 by Ed Katibah (Spatial Program Manager at Microsoft), and aims to make data management self-tuning, self-organizing, and self-maintaining with the development of SQL Server AlwaysOn technologies, to provide near-zero downtime. SQL Server 2008 also includes support for structured and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as BLOBs (binary large objects), but they are generic bit streams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them.

* TABLE
* QUERY
* FORM
* REPORT
* MACRO

**TABLE:**

A database is a collection of data about a specific topic.

***VIEWS OF TABLE:***

We can work with a table in two types,

* Design View
* Datasheet View

***Design View***

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

***Datasheet View***

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot(it cannot be edited).Each time we run query, we get latest information in the dynaset.Access either displays the dynaset or snapshot for us to view or perform an action on it ,such as deleting or updating.

**FORMS:**

A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

***VIEWS OF FORM:***

We can work with forms in several primarily there are two views:

* Design View
* Form View

***Design View:***

To build or modify the structure of a form, we work in forms design view. We can add control to the form that are bound to fields in a table or query, includes textboxes, option buttons, graphs and pictures.

***Form View:***

The form view which display the whole design of the form.

**REPORT:**

A report is used to vies and print information from the database. The report can ground records into many levels and compute totals and average by checking values from many records at once. Also the report is attractive and distinctive because we have control over the size and appearance of it.

**MACRO:**

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report .We write macros to automate the common tasks the work easy and save the time

**Features of SQL- Server**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**About SQL:**

It is also pronounced as Sequel server. This is a client-server database management system developed by Microsoft for large scale databases. It comes into RDBMS (Relational Database Management System) category. It works based on SQL (System query language), a latest and advanced database language.

**Authentication**

A user need to login to this software system in-order to manage the database. SQL server allows us to logon in two ways.

1. Windows authentication
2. SQL Server authentication

In windows authentication, system recognizes the user through his/her windows login credentials. That means, if a user logs into a windows operating system, he/she becomes an authorized user to use all resources that accept windows authentication and available in the system. If the logged in computer is configured as a valid client under a DNS (domain name server), the user becomes eligible to access any resource of any computer that is working under the DNS with widows authentication. In front end (programming) the following connection string can be used to connect to the server.

Server=[server name] ; initial catalog=[database name]; trusted

connection=true;

In SQL server authentication, user will be provided a user id and password. A user from any system in the network can use the user id and password to connect to the database server. In front end (programming) the following connection string can be used to connect to the server.

Server=[server name] ; initial catalog=[database name]; user id=[user id]; password=[password];

**Features of SQL Server**

* License cost is very lower than any RDBMS Systems
* Runs under Windows all windows servers (NT/2000/2003)
* Runs under client versions Windows 95/98/XP … (to install server service, we need to install MSDE – Microsoft SQL Desktop Engine)
* Scalable to meet Enterprise level databases
* Supports data replication
* Supports Data Marts and Data Warehouses
* Provide OLAP service
* English Query tool makes data move available to casual users
* Data transmission services enable easy exchange of data
* Supports distributed transaction
* Centralized management
* Availability of Visual administration tools and wizards
* Generation and transmission of data in XML format

**Advantages of relational databases**

* Redundancy can be reduced
* Inconsistency can avoided
* Irrelevant data can be avoided
* Data can be shared
* Standards can be enforced
* Security restrictions can be applied means access levels can be defined
* Integrity can be maintained
* Conflicting requirements can be balanced

**SQL Server Database objects**

In a SQL server the following database objects can be created and maintained.

* Database Users (user accounts)
* Databases
* Tables
* Views
* Constraints
* Indexes
* Triggers

Stored procedures and functions

4.SYSTEM DESIGN.

**4.1DATA FLOW DIAGRAM**

**Level 0:**

**Doctor**

Login

Managing Patient Healthcare Information

Login

**Administrator**

**Patient**

Doctor Information Management

Login

Review Healthcare

History

Level : 1

Patient Module

**Patient**

Administrator

**Administrator**

Doctor

**Doctor**

Patient History Extract

Extract

**Patient History Extract**

TblPatientData

**4.2ENTITY RELATION DIAGRAM**

User

User

Search

Doctor

VIEW

VIEW

VIEW

4.3 USECASE DIAGRAM.

PATIENT

Administrator

Doctor

4.4 TABLE STRUCTURE:

Table Name: tbl\_admin

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Constraints |
| Username | Nvarchar(30) |  |
| Pass | Nvarchar(30) |  |

Table Name: regis

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Constraints |
| User | Narchar(30) |  |
| Pass | Nvarchar(20) |  |
| Lname | Nvarchar(30) |  |
| Gender | Nvarchar(20) |  |
| Address | Text |  |
| Mno | Nvarchar(30) |  |
| Mail | Nvarchar(20) |  |
| Occup | Nvarchar(30) |  |
| Flag | Nvarchar(30) |  |

Table Name: tbldoctor

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Constraints |
| RecID | Int |  |
| DoctName | Nvarchar(30) |  |
| Dob | Date |  |
| Firstname | Nvarchar(30) |  |
| Lastname | Nvarchar(30) |  |
| Gender | Nvarchar(30) |  |
| Addresss | Text |  |
| Areaname | Nvarchar(30) |  |
| City | Nvarchar(30) |  |
| District | Nvarchar(30) |  |
| Pincode | Nvarchar(20) |  |
| State1 | Nvarchar(30) |  |
| MobileNo | Nvarchar(20) |  |
| Login1 | Nvarchar(30) |  |
| Pass1 | Nvarchar(30) |  |

5.PROJECT IMPLEMENTATION

**5.1 SAMPLE CODING**

Common\_source.cs

using System;

using System.Data;

using System.Configuration;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Net;

using System.Net.Mail;

/// <summary>

/// Summary description for common\_source

/// </summary>

public class common\_source

{

SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["CS"].ConnectionString);

SqlCommand cmd = default(SqlCommand);

SqlDataAdapter ad = default(SqlDataAdapter);

DataSet ds = default(DataSet);

SqlDataReader rd = default(SqlDataReader);

DataTable dt;

int col\_value = 0;

int count1 = 0;

string single\_string;

string multiple\_string;

Boolean flag;

public void insert\_update\_delete(string sql)

{

cmd = new SqlCommand(sql, cn);

conn();

cn.Open();

cmd.ExecuteNonQuery();

cn.Close();

cmd.Dispose();

}

public void conn()

{

if (cn.State == ConnectionState.Open)

{

cn.Close();

}

}

public Boolean record\_availability(string qry)

{

conn();

cn.Open();

cmd = new SqlCommand(qry, cn);

rd = cmd.ExecuteReader();

if (rd.Read())

{

flag = true;

}

else

{

flag = false;

}

cn.Close();

cmd.Dispose();

rd.Close();

return flag;

}

public DataSet bnd(string qry)

{

conn();

ad = new SqlDataAdapter(qry, cn);

ds = new DataSet();

ad.Fill(ds);

cn.Close();

return ds;

}

public int auto\_gen(string field\_name)

{

ds = new DataSet();

ad = new SqlDataAdapter("select " + field\_name + " from Tbl\_AutoGen ", cn);

ad.Fill(ds);

dt = ds.Tables[0];

foreach (DataRow dr in dt.Rows)

{

col\_value = Convert.ToInt16(dr[0]);

}

ds.Dispose();

dt.Dispose();

return col\_value;

}

public void Clear(Control Parent)

{

foreach (Control Child in Parent.Controls)

{

if (Child.HasControls())

Clear(Child);

else

{

if (Child.GetType() == typeof(TextBox))

((TextBox)Child).Text = "";

}

}

}

public int return\_single\_Integer(string qry)

{

conn();

cn.Open();

cmd = new SqlCommand(qry, cn);

int re = Convert.ToInt32(cmd.ExecuteScalar());

return re;

}

public int record\_count(string qry)

{

conn();

DataTable dt1;

DataSet ds1;

ds1 = new DataSet();

ds1 = bnd(qry);

if (ds1.Tables[0].Rows[0].IsNull(0))

{

count1 = 0;

}

else

{

dt1 = ds1.Tables[0];

foreach (DataRow dr in dt1.Rows)

{

count1 = Convert.ToInt16(dr[0]);

}

}

return count1;

}

public float float\_record\_count(string qry)

{

float dec1 = 0;

conn();

DataTable dt1;

DataSet ds1;

ds1 = new DataSet();

ds1 = bnd(qry);

if (ds1.Tables[0].Rows[0].IsNull(0))

{

count1 = 0;

}

else

{

dt1 = ds1.Tables[0];

foreach (DataRow dr in dt1.Rows)

{

dec1 = Convert.ToInt32(dr[0]);

}

}

return dec1;

}

public string return\_single\_string(string qry)

{

conn();

cn.Open();

cmd = new SqlCommand(qry, cn);

single\_string = Convert.ToString(cmd.ExecuteScalar());

return single\_string;

}

public string return\_multiple\_string(string qry)

{

conn();

cn.Open();

cmd = new SqlCommand(qry, cn);

multiple\_string = Convert.ToString(cmd.ExecuteReader());

return multiple\_string;

}

public void Combo\_Bind(DropDownList Obj1, string Qry3, string ColName)

{

Obj1.DataSource = bnd(Qry3);

Obj1.DataTextField = ColName;

Obj1.DataBind();

Obj1.Items.Insert(0, "--Select--");

}

public void MessageBox(string msg, Page frmname)

{

Label lbl = new Label();

lbl.Text = "<script language='javascript'>" + Environment.NewLine + "window.alert('" + msg + "')</script>";

frmname.Controls.Add(lbl);

}

public void DataList\_Bind(DataList Obj2, string SQL\_Query)

{

Obj2.DataSource = bnd(SQL\_Query);

Obj2.DataBind();

}

public void AutoGen\_Update(string colname, int value)

{

string Qry1 = "Update Tbl\_AutoGen set " + colname + " = " + value;

insert\_update\_delete(Qry1);

}

public common\_source()

{

//

// TODO: Add constructor logic here

//

}

public void Mail\_Transfer(string from1, string pass, string to1, string subj1, string mess1)

{

string from = from1; //Replace this with your own correct Gmail Address

string pass1 = pass;

string to = to1; //Replace this with the Email Address to whom you want to send the mail

string subject = subj1;

string message = mess1;

MailMessage mail = new MailMessage();

mail.To.Add(to);

mail.From = new MailAddress(from, "RAIL PREDICTION :: MAIL SERVICE", System.Text.Encoding.UTF8);

mail.Subject = subject;

mail.SubjectEncoding = System.Text.Encoding.UTF8;

mail.Body = message;

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

mail.IsBodyHtml = true;

mail.Priority = MailPriority.High;

SmtpClient client = new SmtpClient();

//Add the Creddentials- use your own email id and password

client.Credentials = new System.Net.NetworkCredential(from, pass1);

client.Port = 587; // Gmail works on this port

client.Host = "smtp.gmail.com";

client.EnableSsl = true; //Gmail works on Server Secured Layer

try

{

client.Send(mail);

//lbluse.Text = "Successfully Mail Sent!!";

//cls1.Clear(this);

}

catch (Exception ex)

{

Exception ex2 = ex;

string errorMessage = string.Empty;

while (ex2 != null)

{

errorMessage += ex2.ToString();

ex2 = ex2.InnerException;

}

HttpContext.Current.Response.Write(errorMessage);

}

}

public void Dial2SMS(string MobileNo, string Message)

{

string url = "http://hpsms.dial4sms.com/api/web2sms.php?username=sofsms&password=Admin@14&to=" + MobileNo + "&sender=SOFSMS&message=" + Message;

WebRequest request = WebRequest.Create(url);

// If required by the server, set the credentials.

request.Credentials = CredentialCache.DefaultCredentials;

// Get the response.

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

}

}

**Default.aspx**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Welcome to Blood Bank</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width; initial-scale=1.0">

<link rel="icon" href="images/favicon.ico" type="image/x-icon">

<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />

<link rel="stylesheet" type="text/css" media="screen" href="css/style.css">

<script type="text/javascript" src="js/jquery-1.8.1.js"></script>

<script type="text/javascript" src="js/superfish.js"></script>

<script type="text/javascript" src="js/jquery.responsivemenu.js"></script>

<script type="text/javascript" src="js/jquery.mobilemenu.js"></script>

<script type="text/javascript" src="js/jquery.flexslider.js"></script>

<script type="text/javascript" src="js/jquery.easing.1.3.js"></script>

<script src="js/script.js"></script>

<script type="text/javascript">

$(window).load(function() {

jQuery('.flexslider').flexslider({

animation: "slide",

slideshow: true,

slideshowSpeed: 7000,

animationDuration: 600,

prevText: "",

nextText: "",

controlNav: true

})

});

</script>

</head>

<body>

<div class="main">

<div class="container\_12">

<!--==============================header=================================-->

<header>

<div class="wrapper">

<h1><a href="index.html"><img src="images/logo.png" alt=""></a><span>roper Business Consulting</span></h1>

<div class="social">

<a href="#" title="Facebook" id="soc1"></a>

<a href="#" title="Twitter" id="soc2"></a>

<a href="#" title="Google +" id="soc3"></a>

<a href="#" title="RSS" id="soc4"></a>

</div>

</div>

<nav>

<ul class="sf-menu">

<li class="current"><strong></strong><a href="index.php">Home</a></li>

<li><strong></strong><a href="FrmDonor.php">Donor Form</a></li>

<li><strong></strong><a href="FrmRequestForm.php">Request Form</a></li>

<li><strong></strong><a href="FrmClients.php">Clients</a></li>

<li><strong></strong><a href="FrmSearch.php">Search</a></li>

<li><strong></strong><a href="FrmAdmin.php">Administrator</a></li>

</ul>

<div class="clear"></div>

</nav>

</header>

<!--==============================content================================-->

<div class="main-slider">

<div class="flexslider clearfix">

<ul class="slides">

<li><img alt="" src="images/slide.jpg">

<div class="flex-caption">

</div>

</li>

<li><img alt="" src="images/slide-1.jpg">

<div class="flex-caption">

</div>

</li>

<li><img alt="" src="images/slide-2.jpg">

</li>

</ul>

</div>

</div>

<section id="content">

<div class="wrapper">

<article class="grid\_12">

<div class="indent">

<div class="extra-wrap">

<h3>Management Consulting</h3>

</div>

<p>Lorem ipsum dolor sit amet, conse etur adipiscing elit. Duis vel nisifes. Vestibulum ullamcorper.

Duis vel nisifes. Vestibulum ullamcorper. Duis vel nisifes. Vestibulum ullamcorper. Duis vel nisifes. Vestibulum ullamcorper.

Duis vel nisifes. Vestibulum ullamcorper. Duis vel nisifes. Vestibulum ullamcorper. Duis vel nisifes. Vestibulum ullamcorper.

</p>

</div>

</article>

<div class="clear"></div>

</div><!--///wrapper-->

</section>

</div>

</div>

</body>

</html>

**5.2 SAMPLE OUTPUT**

7.CONCLUSION

**7.1 BENEFITS**

The Developed system is flexible and robust. This software provides the clarity in all its operation in terms of what, why, who, when, where, how and also reduces the cost of operation. In performing analysis, design, coding and Implementation we have made success in matching the system objectives with the goals of the organization. The newly developed system consumes less processing time. Since the Navigation is maintained throughout the system, they are much user-friendlier and serve the organization purpose. As we done the initial investigation, now we can say that this application possible to create. But as project will progress there may some change in functionality of the project.

**7.2 FUTURE ENHANCEMENT**

The newly developed system, in its present form, is eminently suited to the existing needs. But in order to meet the future needs, which can become progressively more complex the efficiency of the system can be improved by making some simple modifications in the programs.

* Introducing Mobile Apps
* Database may be available in future for long times and information may be use anytime.
* Manage & backup versions of documents online.

8. BIBLIOGRAPHY

**i) Books referred:**

* ASP.NET

1. The Complete Reference by Matthew MacDonald.

* ASP.NET

1. Developers Guide by Buczek

* Software Engineering

1. Roger S. Pressman.

* ASP.NET

1. ASP.NET Professional by Wrox

* C#

1. ASP.NET Professional by Wrox

**ii)Websites:**

[www.c-sharpcorner.com](http://www.c-sharpcorner.com)

[www.asp.net](http://www.asp.net)