E-Health Example

Infrastructure of the E-Health

ABSTRACT

This project is based on the e-health which provides medical advice, latest news on diseases general information, registration of patient, patient records maintaining ,telephone helpline, health news by e-mail, patient health observation, list of hospitals & searching for particular hospital and doctor, before and after comparisons, providing health demos. Providing tips for health and food. The main aim is to provide full detail information and support regarding any disease to patient. It is the medium between a doctor and a patient. The main goal of the project is to give User friendly features. It is a web application which shows and helps patients to collect most of the information about Hospitality and Medical Services. It can used by the patients to view the list of doctors available in their cities and take appointments of doctors with the help of this site. The site will include different articles from famous doctors through e-mail facility. There will be online help for and counseling etc... patients through e-mail facility. * It is maintained by an administrator. * User accounts for patients. The data is well protected for personal use and makes the data processing very fast. * Receive emails from doctors and specialists

about diseases, preventive measures and general awareness. * User should be able to search the records for doctors, patients and related medication information, tips for the health and diet along with the user record. * Unique platform to maintain transparency and accuracy along with formatted order with proper maintenance of data records. * Health information technology plays an important role in supporting decision making, health care delivery system , and management of health services. Many socio-technical factors affect physicians adoption and implementation of health information systems. * Place to post ads. * As it is a web-enabled project low cost and time of project deployment and maintenance.

Chapter - 1

1. INTRODUCTION TO PROJECT

In the present e health is maintained through the storing the information into a single system and then using the search engines to search the data which user wants to. In the current scenario maintaining the search engines and maintaining the data costs a lot and can be feasible if data is in huge volumes. It cannot be feasible for small volumes of data. Data management is enough and it will be a best feasible solution if the data is stored in records and fetched through the simple forms. It will provides medical advice, latest news on diseases general information, registration of patient, patient records maintaining ,telephone helpline, health news by e-mail, patient health observation, list of hospitals & searching for particular hospital and doctor, before and after comparisons, providing health demos. Providing tips for health and food. It is the medium between a doctor and a patient. The main goal of the project is to give User friendly features. It is a web application which shows and helps patients to collect most of the information about Hospitality and Medical Services. It can be used by the patients to view the list of doctors available in their cities and take appointments of doctors with the help of this site. The site will include different articles from famous doctors through e-mail facility. In current system displays pages in continuously but it is very lengthy and very difficult to the user to search what he needed for user easiness here we provide + and - buttons before that heading if he want to see items on that particular pages he can click + button and see the sub items. Finally we will provide 3 dropdown links problem name, hospital name, doctor name and by selecting them we want to give e-mail confirmation regarding appointment. And the patient information also viewed by the doctor and telephone number is used for tele-medication.

Objectives:-

Objectives of this project is to provide the simple web based forms to User with simple interactive forms to maintain transparency, accuracy and simply effective in maintaining the data over recent past years. Here user can register, enter patient related data along with the options to maintain or update the user own records where all those records can be maintained in the database. User can be able to search the records for doctors, patients and related medication information, tips for the health and diet along with the user record. There is no or unique platform to maintain transparency and accuracy along with formatted order with proper maintenance of data records. Data management is very conventional and complicated manner. No existing mechanism to handle this situation in the present market. Just this causing wasting unnecessarily time and cost. In the present e health is maintained through the storing the information into a single system and then using the search engines to search the data which user wants to. In the current scenario maintaining the search engines and maintaining the data costs a lot and can be feasible if data is in huge volumes. It cannot be feasible for small volumes of data. Data management is enough and it will be a best feasible solution if the data is stored in records and fetched through the simple forms. As I am using small forms and a simple data base, following SDLC phases. Before this in the primary research I have done with gathering overall objective of my research topic E health. As a secondary research I have collected the sample data to use, creating the functional flows, best ways to implement this.

Chapter - 2

EXISTING SYSTEM

Existing system refers to the system that is being followed till now. Presently all the health functionalities or services are completed manually. That is if a patient want to consult a doctor he can visit their till his chance called or call-up and take appointment in office hours. To know any general information about any epidemics or diseases that causes panic among the people if not rightly informed on right time. This makes the person very difficult. Out Patient and In Patient tickets are distributed directly. The main disadvantage is that there will be lot of difficulty for the patient. So, all these procedures will be a time consuming one. Draw backs of existing system: * Difficult for patients * Time consuming. To avoid all these limitations and make the working more accurately the system needs to be online.

PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides data accuracy and save disc space. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to consume time. The system requires very low system resources and the system will work only in internet connection. In the existing system displays pages in continuously but it is very lengthy and very difficult to the user to search what he needed for user easiness here this application will provide + and - buttons before that heading if he want to see items on that particular pages he can click + button and see the sub-items in Treeview form. In existing system/site he shows time that is not updated continuously but in our application we have implemented timely updates for every second. Finally we have provided 3 dropdown links problem name, hospital name, doctor name and by selecting them system will to give e mail confirmation regarding appointment. And the patient information also viewed by the doctor and telephone number is used for tele-medication.

Advantages of Proposed System :

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features § This website will provide online help for patients and counseling / advices for specialists. § This website helps all the users to view the list of doctors available in their cities § User is provided the option of monitoring the records that he enter earlier and also he can see the required records with the multiplicity of options provided by him. § From each part of the project the user is provided with the links all the way through framing so that he can go from one option of the project to other as per the necessity. This is bound to be easy and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the major concerns of any good project. § This website will help take appointments of doctors for the users. § In the existing system displays pages in continuously but it is very lengthy and very difficult to the user to search what he needed for user easiness here this application will provide + and - buttons before that heading if he want to see items on that particular pages he can click + button and see the sub-items in Treeview form.

Chapter - 3

3. FEASIBILITY STUDY

Feasibility study is about the viability of a system. The proposed system has to be examined for its technical, economical and operational feasibility. This system for posting news and working the website was inspected with all these aspects in mind.

3.1.1 Technical Feasibility

It is the process of assessing the development internet websites ability to construct a proposed system. Test is made to see whether reliable hardware and software, technical resources capable of meeting the needs of a proposed system can be acquired or developed by webpage in the required time. In this process, since Dot.Net is used for developing the simple internet application, it's seemed to be very feasible. While accessing the technical feasibility, the various issues that are considered are system performance, system interfaces, development processes, risks, failure immunity and security. This system is proven to be technically feasible.

3.1.2 Economic Feasibility

It is a process of identifying the airfare search benefits with a development project. This project is found to be economically feasible since security is the need of the time. The search benefit analysis is made considering the intricacies such as time to considerations, get more details, effectiveness, and maintainable design.

3.1.3 Operational Feasibility

Operational feasibility study is a must, because it ensures that the project implemented in the web pages works the feasibility should be high. The operational feasibility of this project is very high as it automates document control and web interface, which is easy and friendly for the user to use it.

3.2. SYSTEM ENVIRONMENT

3.2.1 HARDWARE REQUIREMENTS

Processor : Pentium-IV or higher Hard Memory : 20GB or higher Monitor : 1024 * 768 Resolutions Ram : 1GB

3.2.2 SOFTWARE REQUIREMENTS

Front End : ASP.Net Back End : Microsoft SQL Server 2000 Operating System : Windows XP Language : C#.Net Framework : .Net 2.0

3.2.3 CLIENT SYSTEM REQUIREMENTS

Browsers : IE 7 or above, Firefox 2 or above Flash Player : 8 or above Monitor Resolution : 1024*768 Resolution

3.3 ASP. NET

ASP.NET it is a part of the .NET Framework and also a new platform from Microsoft for creating applications that are highly distributed across the Internet. Highly distributed means the components of the application, as well as the data, may reside anywhere on the Internet rather than all being contained inside the one software program somewhere. Each part of an application can be referred and accessed by using a standard procedure. ASP.NET is the part that provides the features easily tie up all this capability together for coherent web-based applications. It is a programming framework, but there is a primary difference between this and traditional ASP it uses Common Language Runtime (CLR) capable of running compiled code on a web server to deploy powerful wed-based applications. ASP.NET still use HTTP to communicate to the browser and back, but it can brings added functionality that makes the communication process much richer. If any files have the appropriate extension or contain code then the server routes those files to ASP.NET for processing prior to sending them out to the client. The script or code is then processed and the appropriate content is generated for transmission back to the browser/client because the processing takes place before the results are delivered to the user and also all manner of functionality can be built-in such as database access, component usage and the ordinary programmatic functionality available with scripting languages. ASP.NET applications can be coded by using the plain text edited such as notepad, although this is not the most proficient method to use. Developing of all the other resources that might be required for a particular ASP.NET application, particularly for the user interface, may involve range of particular tools including image-editing programs and HTML editors. 1. ASP.NET also enables you to separate HTML design from the data retrieval mechanism. Therefore changing the HTML deign does not affect the program that retrieve data from the databases. Similarly, server-side scripting ensures that changing data sources does not require a change in HTML documents. 2. ASP.NET has a number of advance features that help you develop robust web applications. The advance features of ASP.NET are based on the .NET Framework.

3.3.1 ASP.NET in .NET Framework

ASP.NET, which is the .NET version of ASP, is built on Microsoft .NET Framework. Microsoft introduced the .NET Framework to help developers create globally distributed software with Internet functionality and interoperability.ASP.NET application include WEB Forms, configuration files and XML, web service files. ASP.NET has a number of advance features that help you develop robust web applications. The advance features of ASP.NET are based on the .NET Framework.

3.3.2 FEATURES OF ASP.NET

1. Compiled Code - Code written in ASP.NET is compiled and not interpreted. It makes ASP.NET applications much faster to execute than other server- side scripts that are interpreted such as scripts written in a previous ASP. 2. Enriched Tool Support - The ASP.NET Framework is provided with a rich toolbox and designer in VS.NET IDE.Some of the features of this powerful tools are drag-and-drop server controls and automatic deployment. 3. Power and Flexibility - The ASP.NET applications are based on the Common Language Runtime (CLR). 4. Simplicity - ASP.NET enables you to build user interfaces that separates application logic from the presentation content. In addition, CLR simplify the application development by using managed code services, such as garbage collection and automatic reference counting. 5. Manageability - The ASP.NET allow you to manage Web applications by storing the configuration information in an XML file . You can also open the XML file in the visual Studio .NET IDE. 6. Scalability - ASP.NET has been designed with scalability and it has some features that helps you to improve performance in a multiprocessor environment. 7. Security - ASP.NET has a options for implementing security and restricting the user access to a web application. All these options are cond within the same configuration file.

3.3.3 ASP.NET Architecture

ASP.NET is based on the fundamental architecture of the .NET Framework. Visual Studio provides a standardized way to combine the various features of this Architecture.

Architechture of Asp.Net

Architecture is explained form bottom to top in the following discussion. At the bottom of the Architecture is Common Language Runtime(CLR) .NET Framework CLR resides on the top of the operating system services. The common language runtime masses and executes the code that targets the runtime. This code called as managed code. The runtime gives an example that is ability for cross language integration. .NET Framework provides set of class libraries. These classes include in base classes, like input ,output and networking classes . The ADO.NET is Microsoft's ActiveX Data Object (ADO) model for .NET Framework. The 4th layer of the framework consists two types of applications they are Windows application model and in parallel Web application model. The Web application model presents ASP.NET it includes Web services and Web forms. ASP.NET comes with built-in control Web Forms which are responsible for generating the user interface(UI). One of the important themes of .NET is association and interoperability between different programming languages. In order to achieve this certain rules must be laid and all the languages must be following these rules. In other words we don't have languages to running around their own extensions and their own new data types. CLS is the collection of the rules and constraints that every language (that seeks to achieve .NET compatibility) should follow. In general the CLR and the .NET Frameworks are designed in such a way that the code has been written in one language we cannot badly used by another language. Hence ASP.NET can be coded in any of the .NET compatible languages whether it is VB.NET, C#(C SHARP), Managed C++ or JavaScript.NET.

3.4 C#.Net

C# ("C-Sharp") is one of the object-oriented programming languages developed by Microsoft. "C# is a modern, object-oriented language it enables programmers to quickly build a wide range of applications for the new Microsoft .NET. Which provides tools and services that are fully exploited in both computing and communications? C# original called codename is "Cool” being released as a beta in 2000. After that Microsoft released different version of language including the latest release of C# 2.0. Some of the basic features of the C# programming language they are namespaces, type-safe variables, multi-dimensional arrays, jagged arrays, operator overloading, indexers, delegates, versioning, attributes and overriding. C# also have two types of parameters they are "pass by reference" and "pass by value" and also have xml based documentation with some special comment tags, Integration with COM components are developed using Visual Studio 2005.

3.4.1 FEATURES OF C#

v There are no pointers used in C#. v In C# Unsafe operations are not allowed like direct memory manipulation. v In C# we don't use “:” or "->" operators. v C# based on the current trend it is very powerful and simple for construct robust applications. v C# includes built in support to turn any component into a web service that can be invoked over the Internet from any application running on any platform. v C# supports encapsulation, polymorphism and interfaces. v In C# we cannot convert double to a Boolean. v C# supports the COM and windows based applications.

3.4.2 SQL Server

SQL - Structured query language.

3.4.2.1 INTRODUCTION TO SQL SERVER:

To create a database that determines the name of the user (who creates the database) and database size. Then all file groups are used to store it and retrieve it. Before creating a database we must follow these steps: * Take Permission to create a database defaults to members of the system administrator and Database Creator has some fixed server roles, although they can grant permissions to other users. * User who creates their own database becomes owner of the database. * We can create maximum of 32,767 databases on a server. * Name of the database must be follows rules for identifiers. Three types of files are used to store a database:

· Primary files

Primary files contain the startup information for the database. It can also used for stores the data. one primary file allocated to one database.

· Secondary files

Secondary files hold all the data that data is not fit into the primary data file. In Databases primary file cannot hold all the data in that situation we use secondary file. Some databases have large data files so we need to use secondary files or some databases may use secondary files on separate disk drives to spread the data into multiple disks.

· Transaction log

Log files are used to hold the log information and to recover the database. Each database contains one log file although there may be more than one log file. Minimum size the log file is 512 kilobytes (KB). It specifies a maximum size to which the file is permitted to grow. This prevents the file is growing data is added until the disk space is exhausted. To specify a maximum size of file we used to write MAXSIZE parameter.

CREATING DATABASE PLAN:

The first step of the database creation is creating a plan that serves both guide to be used for database implementation and functional specification for the database after that it has been implemented. Detailed database design is dictated by the complexity and size of the database application as well as the user population. Database can vary nature and complexity as well as planning of application. Single person can easily design and use a database or it can be large or complex for example, bank can handle many of transactions at a time. In planning database we use these basic steps: 1. First we need to gather all information. 2. Objects identified. 3. Object model. 4. Types of information for objects. 5. Relationships between the objects.

3.5 SYSTEM DESIGN

3.5.1 DATABASE DESIGN

Database is a collection of interrelated data that data we stored with minimum redundancy to serve for many users quickly and competently. General objective of database design is to make the data access easy, inexpensive and flexible to the user. Database design is used to define and then specify the structure of business used in the client/server system. A business object is nothing but information that is visible to the users of the system. The database must be normalized one.

Data Normalization

The entities along with their attributes can be stored in many different ways into a set of tables. The methods of arranging these attributes are called normal forms. The theory behind the arrangement of attributes into table is known as normalization theory. It helps in, * Minimization of duplication data. * Providing flexibility to support different functional requirements. * Enabling the model to be translated to database design. All relations in a relational database are required to satisfy the following condition, every value in a relation each attribute value is atomic so far as system is concerned. Advantages of normalization are: * Helps in reduction in the complexity of maintaining data integrity by removing the redundant data. * It reduces inconsistency of data

First normal form:

* Eliminate the repeating fields. * Creates a row for each occurrence of a repeated field * Allows exploitation of column functions

Second Normal Form:

The second normal form has the characteristics of the first normal form and all the attributes must fully be dependent on the primary key.

Input Design:

Input design is process of converting the user-oriented inputs to the computer-based format. Goal of the designing input data is to make automation as easy and easily finds errors as possible. Provides good input design for the application easy data input and selected features are adopted. There is some basic requirements of the input design such as user friendliness, consistent format and give right message for interactive dialogue and it also helps the user at right time are also considered for development of the project. Following points are consider while designing the input: * Data to input? * Medium to use? * How data is arranged or coded? * Input provides dialogue to the users. * To detect the errors we need validation for data items and transactions. * when Methods for performing input validation and steps to follow when errors occur. Minimize the number of input actions required from user. This can be accomplished by using the mouse to select from predefined set of inputs. In application the user can select the options by using the mouse. The user is allowed to choose priority, mode of transport using predefined set of values. Maintain consistency between information display and data input. The visual characteristics of the display (e.g. text size, color etc) should be carried over to the input domain. In this project the status information are represented by different colors. Allow the user to customize input. An expert user might decide to create customer commands or dispense with some types of warning messages and action verification.

Output Design:

When designing output, systems analyst must accomplish the following: * Determine what information to present * Decide whether to display, print the information and select the output medium * Arrange the presentation of information in an acceptable format * Decide how to distribute the output to intended recipient Accomplishing the general activities listed above will require specific decisions, such as whether to use preprinted forms when preparing reports and documents, how many line to plan on printed page, or whether to user graphics and color. The output design is specified on layout forms, sheets that describe the location characteristics (such as length and type), and format of the column headings and pagination

Table Name: Login

This table stores the details about the login .User login table contains the fields username and password. Field Name Data Type Description User Name Text User Name Password Text User Name

Table Name: Admin Registration

This table stores the details about the admin .Admin registration table contains the fields username and password. Field Name Data Type Description User Name Text User Name Password Text User Name

Table Name: User Registration

This table stores the details about the user registration. The User Registration table contains the fields. Field Name Data Type Description Name Text User Name Age Integer Age Gender Text Gender Address Text Address Phone no Long Phone Number User Name Long User Name Password Long Password

Table Name: Doctor Registration

This table stores the details about the doctor registration. The Doctor Registration table contains the fields. Field Name Data Type Description Dr name Text Doctor Name Gender Text Gender Dept Text Department Consultation time Integer Consultation Time Address Text Address Contact no Long Contact Number Emergency no Long Emergency Number Consult fee Long Consultation fees Date Date/Time Date Consultation day Text Consultation Day Tot pat Double Total Patient

3.6 DATA FLOW DIAGRAM:

Data flow oriented techniques advocate that the major data items handled by a system must be first identified and then the processing required on these data items to produce the desired outputs should be determined. The DFD (also called as bubble chart) is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on these data, and the output generated by the system. It was introduced by De Macro (1978), Gene and Sarson (1979).The primitive symbols used for constructing DFD's are:

Symbols used in DFD:

A circle represents a process. A rectangle represents external entity A square defines source of the system data. An arrow will identify the dataflow. Double line with one end closed indicates data store

3.6.1CONTEXT DIAGRAM:

3.6.1.1Level 1 DFD Administrator:

3.6.1.2 Level 1 DFD User

3.6.1.3 Level 2 DFD User:

3.6.1.4 Level 2 DFD Admin:

3.8 SYSTEM TESTING:

Testing is a process to show the correctness of the program. Testing is needed to show completeness, t improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time. Testing software extends throughout the coding phase and it signify the ultimate review of the design ,configuration and coding. Based on the way the software reacts to these testing, we can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

Testing involves

* Unit testing * Integration testing * Acceptance testing The first level of test is unit testing. The purpose of unit testing is to ensure that each program is fully tested. The second step is integration testing. In this individual program units or programs are integrated and tested as a complete system to ensure that the software requirements are met. Acceptance Testing involves planning and the execution of various types of tests in order to demonstrate that the implemented software system satisfies the requirements. Finally our project meets the requirements after going through all the levels of testing.

Chapter - 4

4.IMPLEMENTATION AND MAINTENANCE

4.1System Implementation:

Implementation includes all those activities that take place to convert from the old system to the new system. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Sometimes successful implementation may not guarantee any improvement in the organization using the new system. The implementation phase includes the following tasks: * Careful planning. * Investigation of the system and constraints. * Design of methods to achive the change over phase. * Training of staff in the change over phase. * Evaluation of change over. The method of implementation and time scale to be adapted is found out initially.Next,the system is tested properly and at the same time the users were trained in the new environment. In Infrastructure of the E-Health implementation process is successfully implemented the system by satisfying all the aspects of the user. All the procedure are designed to minimise the users resistance to change and make a attitude for full utilization of the system.

4.2 MODULE DESCRIPTION:

1. Admin module. 2. Client module. 3. Visitor Module

4.2.1 ADMIN MODULE

Administrator can control all processes of the entire project. Admin can login at any time to see the process of all users. Admin can monitor all the activities of the user. Admin can upload banner, text or HML ads, representing products or forms in the system. Administrator has the following functions: Ø Can add / update / delete Patients upon request or patients client directly register to be part of the system. Ø Can add / update / delete Doctors upon request or patients client directly register to be part of the system. Ø Can add / update / delete Diseases info directly to HTML static pages and through webpage to dynamically display data in treeview form. Ø Send E-Mails to users about diseases info whenever required and this will ensure that users will have update to date data ontime without searching outside.

4.2.2 CLIENT MODULE

Clients are either doctor / patients can navigate to all static pages and dynamic pages that are specific to the particular users in the project. Example dr can reply emails to particular patient for any particular disease treatment/advice. Clients can login at any time to see his process. They will receive emails and can read only posts that are specific to particular user. Clients does not require any extra softwares or hardwares except the system with internet connection and client enabled browser. Clients has the following functions: Ø Can register and come a client, it may be doctor or patient. Ø Can add / update / delete personal details except the user id as it is uniqure for every user. Ø Can search for doctor and hospitals based on treatment required or disease as input. Ø Receive emails from doctors for any disease treatment or advices from specialists for any disease.

4.2.3 VISITOR MODULE:

Visitors are not registered users who visit only rarely or once to read or gather any information. These users will not receive any mails or have oportunity to either search for doctor / hospitals. They can navigate to all static pages and can contact admins for displaying their ads in ad banners. Visitors can become users by registering online. Even visitors also does not require any extra softwares or hardwares except the system with internet connection and client enabled browser. Clients has the following functions: Ø Can navigate to all static pages, can read pages that contain info about diseases or can gather information of general purpose. Ø Can contact admin for posting ads in the site and can become registered user by registering. On overall, Infrastructure of the E-Health has following features. 1. Login authentication required for all users. 2. Registering the patients and doctors. 3. Updating the details of the patients and doctors. 4. Send and receive emails to all users. 5. Updating the personal details of users. 6. Search for doctors / hospitals. 7. Treeview of the diseases info can expand and read only particular disease info that user wants to read. We have implemented N-Tier architecture in this project.

We did not accessed database from the UI pages. Project has a data layer class which performs all the database related tasks. We have added connecting string in the web.config file and accessed in this layers. This will help us support or migrate to another database back end easily. Passwords are Encrypted (or) Decrypted at the time of registration and login. User id will be checked for availability and is provided to the User if it is available or user will get instant message as ad is unavailable and have to use an alternate ID. The login is then authenticated while login. Once the user fills the minimum data required for registration, he becomes a registered user of the website. Now he can login to the Home Page with his user ID and password. He must know the username and password. Nobody can enter into the login form. The Username and password should be kept secrecy by the User. As the password is encrypted and stored it is more secured. During login, if the user ID is wrongly entered, then the user is provided with an error page. If the password is wrongly entered, then he is provided with the same login box. If the login user is Administrator, he is provided with a home page. Once the login is successful every user is redirected to their respected Workflows through which they can navigate to any screens related or assigned to particular user. Following is the code snippet for Registration page:

Registration.aspx:

Registration required controls are placed in the page and html code is added which is used throughout for common look and feel. <%@ Page Language="C#" AutoEventWireup="true" CodeFile="Registration.aspx.cs" Inherits="Registration" %> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "https://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="https://www.w3.org/1999/xhtml" > <head id="Head1" runat="server"> <title>Home</title> <link href="style.css" rel="stylesheet" type="text/css" /> <script type="text/javascript" src="js/jquery-1.1.3.1.min.js"></script> <script type="text/javascript" src="js/jquery.easing.min.js"></script> <script type="text/javascript" > function validate() { var digits="0123456789"; var temp; var pinCde=document.getElementById("<%=txtPinCode.ClientID%>"); for (var i=0;i<pinCde.value.length;i++) { temp=pinCde.value.substring(i,i+1); if (digits.indexOf(temp)==-1) { alert("Invalid Pincode. Please try again."); pinCde.focus(); return false; } } var reEmail = /^(".*"|[A-Za-z]w*)@([d{1,3}(.d{1,3}){3}]|[A-Za-z]w*(.[A-Za-z]w*)+)$/; var

```
emailId=document.getElementById("<%=txtEmailId.ClientID %>"); if (emailId.value.match(reEmail) == null) {
alert("Your email address seems incorrect. Please try again."); emailId.focus(); return false; } var
contactNo=document.getElementById("<%=txtContactNo.ClientID%>"); for (var i=0;i<contactNo.value.length;i++) {
temp=contactNo.value.substring(i,i+1); if (digits.indexOf(temp)==-1) { alert("Invalid Contact No. Please try again.");
contactNo.focus(); return false; } } } </script> </head> <body> <form id="form1" runat="server"> <div id="wrap">
<div id="topbg"> </div> <div id="wrap2"> <div id="header"> <div id="headercontent"> </div> </div> <!-- start
menu --> <div id="menu" style="width:99%; background-color:#D2D8E4; margin-left:5px; height:94px;"> <div
id="logo_banner" style="background-color:red; width:180px;height:92px;"/> <div id="header_banner"
style="width:765px; height:92px; margin-left:182px;"> <table id="test" style="width:763px;height:91px;"><tr><td
align="left" valign="top"> <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="../download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,40,0"
width="763px" height="91px" > <param name="movie" value="flash/header.swf"/> <param name="quality"
value="high"/> <param name="menu" value="false"/> <object data="flash/header.swf" width="763px"
height="91px" type="application/x-shockwave-flash"> <param name="quality" value="high"/> <param
name="menu" value="false"/> <param name="pluginurl" value="../www.macromedia.com/go/getflashplayer"/>
FAIL (the browser should render some flash content, not this). </object> </object></td></tr> </table> </div>
</div> </div> <!-- end menu --> <!-- start banner --> <div id="banner" style="width:99%; background-
color:#D2D8E4; margin-left:5px; height:275px;"> <table id="tblBanner" style="width:937px;height:270px;"><tr><td
align="left" valign="top"> <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="../download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,40,0"
width="300px" height="270px" > <param name="movie" value="flash/logo.swf"/> <param name="quality"
value="high"/> <param name="menu" value="false"/> <object data="flash/logo.swf" width="300px"
height="270px" type="application/x-shockwave-flash"> <param name="quality" value="high"/> <param
name="menu" value="false"/> <param name="pluginurl" value="../www.macromedia.com/go/getflashplayer"/>
FAIL (the browser should render some flash content, not this) </object> </object></td> <td align="left"
valign="top"> <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="../download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,40,0"
```

```
width="637px" height="270px" > <param name="movie" value="flash/slide_show.swf"/> <param name="quality"
value="high"/> <param name="menu" value="false"/> <object data="flash/slide_show.swf" width="637px"
height="270px" type="application/x-shockwave-flash"> <param name="quality" value="high"/> <param
name="menu" value="false"/> <param name="pluginurl" value="../www.macromedia.com/go/getflashplayer"/>
FAIL (the browser should render some flash content, not this) </object> </object></td></tr> </table> </div> <!--
end banner --> <!-- Start content --> <div id="content"> <div id="left"> <div class="post"> <div class="postheader">
</div> <div class="postcontent" style="height:330px"> <h2><a href="#">Registration</a></h2> <div
style="height:339px; width: 595px; position: absolute; border-top-width: 1px; border-left-width: 1px; border-left-
color: silver; border-bottom-width: 1px; border-bottom-color: silver; border-top-color: silver; border-right-width:
1px; border-right-color: silver;"> <div style="left: 11px; width: 575px; position: absolute; top: 9px; height: 292px;
border-right: silver 1px solid; border-top: silver 1px solid; border-left: silver 1px solid; border-bottom: silver 1px
solid;"> &nbsp; &nbsp;&nbsp;&nbsp; <asp:TextBox ID="txtState" runat="server" Style="z-index: 103; left: 373px;
position: absolute; top: 101px" MaxLength="50"></asp:TextBox> <asp:TextBox ID="txtName" runat="server"
Style="z-index: 104; left: 75px; position: absolute; top: 27px" Width="446px" MaxLength="50"></asp:TextBox>
<asp:TextBox ID="txtAddress" runat="server" Style="z-index: 105; left: 76px; position: absolute; top: 53px"
Rows="1" TextMode="MultiLine" Width="446px" Height="38px" MaxLength="100"></asp:TextBox> <asp:TextBox
ID="txtCity" runat="server" Style="z-index: 106; left: 76px; position: absolute; top: 101px"
MaxLength="50"></asp:TextBox> <asp:RequiredFieldValidator ID="rfvCity" runat="server"
ControlToValidate="txtCity" Display="Dynamic" ErrorMessage="*" Style="z-index: 107; left: 233px; position:
absolute; top: 101px" Width="10px" ValidationGroup="Registration"></asp:RequiredFieldValidator> <asp:TextBox
ID="txtUserId" runat="server" MaxLength="50" Style="z-index: 106; left: 76px; position: absolute; top:
188px"></asp:TextBox> <asp:RequiredFieldValidator ID="rfvUser" runat="server" ControlToValidate="txtUserId"
Display="Dynamic" ErrorMessage="*" Style="z-index: 107; left: 235px; position: absolute; top: 193px"
ValidationGroup="Registration" Width="10px"></asp:RequiredFieldValidator> <asp:Label ID="Label3"
runat="server" CssClass="descAdminLabel" Style="z-index: 151; left: 3px; position: absolute; top: 101px"
Text="*City:"></asp:Label> <asp:TextBox ID="txtPwd" runat="server" Style="z-index: 106; left: 79px; position:
absolute; top: 216px" TextMode="Password" MaxLength="16" Width="146px"></asp:TextBox>
```

```
<asp:RequiredFieldValidator ID="rfvPwd" runat="server" ControlToValidate="txtPwd" Display="Dynamic"
ErrorMessage="*" Style="z-index: 107; left: 233px; position: absolute; top: 219px" ValidationGroup="Registration"
Width="10px"></asp:RequiredFieldValidator> <asp:TextBox ID="txtRetype" runat="server" Style="z-index: 106; left:
373px; position: absolute; top: 217px" TextMode="Password" MaxLength="16"></asp:TextBox>
<asp:CompareValidator ID="cfvPwd" runat="server" ControlToValidate="txtRetype" Display="Dynamic"
ErrorMessage="Password mismatch" Style="z-index: 107; left: 83px; position: absolute; top: 240px"
ValidationGroup="Registration" Width="125px" ControlToCompare="txtPwd"></asp:CompareValidator>
<asp:RequiredFieldValidator ID="rfvRetype" runat="server" ControlToValidate="txtRetype" Display="Dynamic"
ErrorMessage="*" Style="z-index: 107; left: 528px; position: absolute; top: 217px" ValidationGroup="Registration"
Width="10px"></asp:RequiredFieldValidator> <asp:Label ID="Label2" runat="server" CssClass="descAdminLabel"
Style="z-index: 151; left: 3px; position: absolute; top: 219px" Text="*Password:"></asp:Label> <asp:Label
ID="Label1" runat="server" CssClass="descAdminLabel" Style="z-index: 151; left: 6px; position: absolute; top:
194px" Text="*User Id:"></asp:Label> <asp:RequiredFieldValidator ID="rfvState" runat="server"
ControlToValidate="txtState" Display="Dynamic" ErrorMessage="*" Style="z-index: 108; left: 528px; position:
absolute; top: 101px" Width="10px" ValidationGroup="Registration"></asp:RequiredFieldValidator>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtContactNo"
Display="Dynamic" ErrorMessage="*" Style="z-index: 108; left: 528px; position: absolute; top: 153px"
ValidationGroup="Registration" Width="10px"></asp:RequiredFieldValidator> <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server" ControlToValidate="txtEmailId" Display="Dynamic" ErrorMessage="*"
Style="z-index: 108; left: 233px; position: absolute; top: 153px" ValidationGroup="Registration"
Width="10px"></asp:RequiredFieldValidator> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;&nbsp;
<asp:TextBox ID="txtEmailId" runat="server" Style="z-index: 114; left: 76px; position: absolute; top: 153px"
MaxLength="30"></asp:TextBox> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;&nbsp;
<asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName" Display="Dynamic"
ErrorMessage="*" Style="z-index: 119; left: 527px; position: absolute; top: 27px" Width="10px"
ValidationGroup="Registration"></asp:RequiredFieldValidator> <asp:RequiredFieldValidator ID="rfvAddr"
runat="server" ControlToValidate="txtAddress" Display="Dynamic" ErrorMessage="*" Style="z-index: 120; left:
```

```
528px; position: absolute; top: 53px" Width="10px"
ValidationGroup="Registration"></asp:RequiredFieldValidator> &nbsp; &nbsp; &nbsp;&nbsp; <asp:TextBox
ID="txtContactNo" runat="server" Style="z-index: 127; left: 373px; position: absolute; top: 153px"></asp:TextBox>
<asp:TextBox ID="txtPinCode" runat="server" Style="z-index: 128; left: 76px; position: absolute; top:
127px"></asp:TextBox> <asp:Label ID="lblPrsnInfo" runat="server" Font-Bold="True" Style="z-index: 129; left:
237px; position: absolute; top: 4px" Text="USER DETAILS" CssClass="subheadings"></asp:Label> <asp:Label
ID="lblState" runat="server" Style="z-index: 130; left: 282px; position: absolute; top: 101px" Text="*State:"
CssClass="descAdminLabel" ></asp:Label> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; <asp:Label
ID="lblEmailId" runat="server" Style="z-index: 138; left: 3px; position: absolute; top: 153px" Text="*Email Id:"
CssClass="descAdminLabel" ></asp:Label> <asp:Label ID="lblContactNo" runat="server" Style="z-index: 139; left:
282px; position: absolute; top: 153px" Text="*ContactNo:" CssClass="descAdminLabel" ></asp:Label> &nbsp;
&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp; <asp:Label ID="lblName" runat="server" Style="z-index:
144; left: 3px; position: absolute; top: 27px" Text="*Name:" CssClass="descAdminLabel" ></asp:Label> &nbsp;
&nbsp;&nbsp; <asp:Label ID="lblCountry" runat="server" Style="z-index: 146; left: 282px; position: absolute; top:
127px" Text="*Country:" CssClass="descAdminLabel" ></asp:Label> <asp:Label ID="lblPinCode" runat="server"
Style="z-index: 147; left: 3px; position: absolute; top: 127px" Text="&nbsp;&nbsp;PinCode:"
CssClass="descAdminLabel" ></asp:Label> &nbsp; &nbsp; &nbsp; <asp:Label ID="lblCity" runat="server" Style="z-
index: 151; left: 284px; position: absolute; top: 220px" Text="*Re-type:" CssClass="descAdminLabel" ></asp:Label>
&nbsp; <asp:Label ID="lblAddress" runat="server" Height="19px" Style="z-index: 153; left: 3px; position: absolute;
top: 53px" Text="*Address:" CssClass="descAdminLabel" ></asp:Label> &nbsp; &nbsp; <asp:DropDownList
ID="ddCountry" runat="server" Style="z-index: 155; left: 373px; position: absolute; top: 127px" Width="155px">
<asp:ListItem Value="US">United States</asp:ListItem> </asp:DropDownList> <hr style="z-index: 164; left: 26px;
width: 522px; position: absolute; top: 185px" /> &nbsp; <asp:Button ID="btnApply" runat="server" Style="z-index:
159; left: 229px; position: absolute; top: 261px" Text="Apply" Width="70px" ValidationGroup="Registration"
OnClientClick="return validate();" OnClick="btnApply_Click" /> &nbsp; &nbsp;&nbsp; &nbsp; </div> <asp:Label
ID="lblErr" runat="server" Font-Bold="True" ForeColor="Red" Style="z-index: 105; left: 11px; position: absolute;
top: 312px"></asp:Label> </div> </div> <div class="postbottom"> </div> </div> </div> <!-- Start sidebar --> <div
```

id="sidebar"> <h3 style="padding-left:20px;text-decoration:underline;"> Scrolling side bar</h3> </div> <!-- End sidebar --> <div class="clear"></div> </div> <!-- End content --> <!-- start footer --> <div id="footer"> Best viewed in Firefox 2, IE 7, Flash Player 8, 1024x768 screen resolution and above<br/> Copyright &copy; Infrastructure of the E-Health| <a href="#">Home</a> | <a href="#">About Us</a> | <a href="#">Services</a> | <a href="#">Clients</a>| <a href="#">Contacts</a> <div class="credit"><a href="" title="">Designed by </a> by Infrastructure of the E-Health </div> </div> <!-- End footer --> </div> <div id="btmbg"> </div> </form> </body> </html> Code snippet of registration.aspx.cs code behind file: Used separate class files for db connections and password encryption kind of util methods in util file protected void btnApply_Click(object sender, EventArgs e) { try { this.setDistributorsRegProps(); objReg.addRegistrationDetails(objReg); lblErr.Text = ""; if (objReg.ErrChk == "E") { lblErr.Text = "User id already exists."; } else { if (objReg.ErrChk == "S" && objReg.RegId != 0) { //Redirect to registration successful page } else { lblErr.Text = "Registration Failed.Error from DB."; } } } catch (Exception ex) { lblErr.Text = ex.Message.ToString(); } } public void setDistributorsRegProps() { try { objReg.Name = txtName.Text; objReg.Addr = txtAddress.Text; objReg.City = txtCity.Text; objReg.State = txtState.Text; if (txtPinCode.Text != "") { objReg.Pin = Convert.ToInt64(txtPinCode.Text); } else { objReg.Pin = 0; } objReg.Country = ddCountry.SelectedValue.ToString(); objReg.Email = txtEmailId.Text; if (txtContactNo.Text.Trim() != "") { objReg.ContactNo = Convert.ToInt64(txtContactNo.Text); } else { objReg.ContactNo = 0; } Utility utilObj = new Utility(); //Password encryption objReg.Password = utilObj.base64Encode(txtPwd.Text); objReg.UserId = txtUserId.Text; } catch (Exception ex) { throw (ex); } } Supporting class files for registration : Connection strings are placed in constructors public clsRegistration() { conStr = ConfigurationManager.AppSettings["strConn"].ToString(); } Connects to databse and performs necessary actions public void addRegistrationDetails(clsRegistration regObj) { conReg = new SqlConnection(conStr); try { cmdReg = new SqlCommand("up_insert_Registration", conReg); cmdReg.CommandType = CommandType.StoredProcedure; SqlParameter parAddrId = new SqlParameter("@RegId", SqlDbType.Int, 4); parAddrId.Direction = ParameterDirection.Output; SqlParameter rtrnStmt = new SqlParameter("@rtrnStmt", SqlDbType.VarChar, 1); rtrnStmt.Direction = ParameterDirection.Output; cmdReg.Parameters.Add(parAddrId); cmdReg.Parameters.Add(rtrnStmt); cmdReg.Parameters.AddWithValue("@Name", regObj.Name); cmdReg.Parameters.AddWithValue("@Address", regObj.Addr); cmdReg.Parameters.AddWithValue("@City", regObj.City);

cmdReg.Parameters.AddWithValue("@State", regObj.State); if (regObj.Pin == 0) { cmdReg.Parameters.AddWithValue("@Pincode", System.DBNull.Value); } else { cmdReg.Parameters.AddWithValue("@Pincode", regObj.Pin); } cmdReg.Parameters.AddWithValue("@Country", regObj.Country); cmdReg.Parameters.AddWithValue("@Email_Id", regObj.Email); if (regObj.ContactNo == 0) { cmdReg.Parameters.AddWithValue("@Contact_No", System.DBNull.Value); } else { cmdReg.Parameters.AddWithValue("@Contact_No", regObj.ContactNo); } cmdReg.Parameters.AddWithValue("@User_Pwd", regObj.Password); cmdReg.Parameters.AddWithValue("@ClientName", regObj.UserId); conReg.Open(); cmdReg.ExecuteNonQuery(); regObj.ErrChk = cmdReg.Parameters["@rtrnStmt"].Value.ToString(); if (null != cmdReg.Parameters["@RegId"] && cmdReg.Parameters["@RegId"].Value.ToString() != "") { regObj.RegId = Convert.ToInt64(cmdReg.Parameters["@RegId"].Value.ToString()); } } catch (Exception x) { throw (x); } finally { conReg.Close(); } } Utility class file that has password encryption and decryption methods // Encode data public string base64Encode(string sData) { try { byte[] encData_byte = new byte[sData.Length]; encData_byte = System.Text.Encoding.UTF8.GetBytes(sData); string encodedData = Convert.ToBase64String(encData_byte); return encodedData; } catch (Exception ex) { throw new Exception("Error in base64Encode" + ex.Message); } } // Decode data public string base64Decode(string sData) { System.Text.UTF8Encoding encoder = new System.Text.UTF8Encoding(); System.Text.Decoder utf8Decode = encoder.GetDecoder(); byte[] todecode_byte = Convert.FromBase64String(sData); int charCount = utf8Decode.GetCharCount(todecode_byte, 0, todecode_byte.Length); char[] decoded_char = new char[charCount]; utf8Decode.GetChars(todecode_byte, 0, todecode_byte.Length, decoded_char, 0); string result = new String(decoded_char); return result; } Web.config file connection string <configuration> <appSettings> <add key="strConn" value="server=localhost;uid=sa;pwd=XXXX;database=XXXX"/> </appSettings> <connectionStrings/> Used try-catch in data layer to catch all database exceptions. This exception handler will record all exceptions from the database. The details recorded include the name of the command being executed, stored proc name, parameters, connection string used etc. After recording the exception, it is re thrown so that another layer in the application can catch it and displayed in a message label error bar which is displayed under header banner and above body layer.Other than above mentioned code session variables are used and helps to identify users and navigate to other pages accordingly. We have implemented N-Tier architecture in this project. We did not accessed database from the UI pages. Project has a data layer class which performs all the database related tasks. We have added connecting string in the web.config file and accessed in this layers. This will help us support or migrate to another database back end easily.

4.4 System Maintenance:

After the installation phase is completed and the user is adjusted to the changes created by the new system,evaluation and maintenance is to continue to bring the new system to the standards. If the new information is consistent with the design specification, changes have to be made. Maintenance is actually the implementation of the post implementation review plan. Maintenance is necessary to eliminate errors in the working system during its working life and to tune the system to any variation in its working environment.There are also some errors detected that must be corrected.The quality assurance goal is develop a procedure for corecting errors and enhancing the software.This procedure improves quality assurance by encouraging complete reporting and log of problems,ensuring that reported problems are promptly forwarded to the appropriate group of resolution. In our project if any error is reported it must be corrected.

Chapter - 5

5.1 SUMMARY OF IMPLEMENTATIONS:

1. Implemented N-Tier architecture. Code is maintained separately in each layer. UI in .asp, code related to functionality in .cs code-behind file and separate data layer class which performs all the database related tasks. 2. Password encryption and decryption using base64encoder. 3. Proper handling of try, catch and finally blocks. 4. Different utility classes to handle common methods and access it wherever required. 5. Maintained keys of connection settings in web configuration. 6. Did not store heavy content in session variables. 7. Used style sheets to maintain uniformity and ease of changing colors and fonts throughout the application.

5.2 COMPARISON OF RESEARCH FINDINGS WITH IMPLEMENTATION:

1. Multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which the presentation, the application processing, and the data management are logically separate processes. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of "multi-tier architecture" refers to three-tier architecture. The concepts of layer and tier are often used interchangeably. However, one fairly common point of view is that there is indeed a difference, and that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure. Hence as we have implemented n-tier it is easy to make changes to each layer without affecting other layers and prevents code collapse and defects. 2. The Base64 term refers a specific MIME content for transfer encoding and also used as a generic term for any similar encoding scheme that encodes the binary data by treating numerically and translating into a base 64 representation. The particular choice of the base is due to the history of character that sets encoding: one can choose 64 characters that is both part of the subset common to encodings and also printable. This combination leaves the data to be modified in transit through systems, such as emails which were not 8-bit clean. The choice of characters is difficult. Earliest this types of instances encoding were created for dialup communication between systems running the same Operating System - for e.g. uuencode for UNIX, Bin Hex for the TRS-80 - and therefore could make more assumptions about what type of characters were safe to use. For instance, uuencode uses uppercase letters, digits, and many punctuation characters not using the lowercase letters. Sometimes UNIX was used with terminals that did not support the distinct letter case. Unfortunately there is interoperability with non-UNIX systems , punctuation characters do not exist in other traditional character sets. The MIME Base64 encoding replaces the punctuation characters with lowercase letters, reasonable requirement by the time it was designed. In MIME Base64 the first 62 values use uppercase(A-Z),lowercase(a-z), and digits(0-9) . There is some other similar systems derived from Base64, that share this property but different in the symbols chosen for the last two values: example is UTF-7. As we have implemented code encryption / decryption with base64 encoder it will improve security that prevents hacking to much extent. 3. Use try-catch in data layer to catch all database exceptions. This exception handler should record all exceptions from the database. The details

recorded should include the name of the command being executed, stored proc name, parameters, connection string used etc. After recording the exception, it could be re thrown so that another layer in the application can catch it and take appropriate action. We have handled try, catch and finally to catch errors and exceptions in data layer and transferred them to another layer from where this method is called and displayed to user. Finally block is used to perform necessary actions that need to be handled every time even when an exception is occurred. Here we have handled closing database connections. 4. Separate application into multiple assemblies. Group all commonly used utility methods into a separate class. We have implemented commonly used methods in utility class file and moved other necessary methods to user class files that has code related to db connections which can be used while db connections and other methods that are related to User details. 5. Declare connection settings in web configuration. We have implemented connection settings keys - add key in web configuration and this key are used throughout the application and hence it is easy to migrate database. 6. Do not store large objects in session. Storing large objects in session may consume lot of server memory depending on the number of users. We have implemented application using minimum number of session variables and very light data in sessions. We have cleared session variables in invalid conditions or exceptions. 7. Always use style sheet to control the look and feel of the pages. Never specify font name and font size in any of the pages. Use appropriate style class. This will help you to change the UI of your application easily in future. Also, if you like to support customizing the UI for each customer, it is just a matter of developing another style sheet for them. We have implemented styles for almost all controls and used all over the application. Hence it will be easy to change look and feel very easily when required.

5.3 RESULTS AND VIEWS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: - 1. In this project it offer user to enter the data through easy and interactive forms. This is very helpful for client to enter the required information through so much ease. 2. The user is mostly worried about the validity of the data, whatever he was entering. For every new creation there are many checks on each stage, data entry or updating so that user cannot enter the invalid data which can be create trouble later . 3. Sometimes the user finds in the later stages of using project then he needs to be update the data that he entered earlier. There are so many options to him by which he can update the data in the records. And also there is restriction for him that he cannot vary the primary data field .So that it keeps the validity of the data to be longer extent. 4. The User was provided the option of monitoring the records he entered earlier and also he can view the required records with the multiplicity of options provided by the user. 5. From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and user friendly as per user necessity. i.e, we can say that the project is user friendly which is one of the primary concerns of any of the projects done previous. 6. Data storage and retrieval will become quicker and easier to maintain because data has to be stored in a systematic, good manner in a single database. 7. Decision making process must be greatly enhanced because of quicker processing of information since data collection from information available on the computer it takes less time than the manual system. 8. Allocating of sample results become more faster because at a time the user can be able to see the data records of previous years. 9. Easier and faster data transfer through the most recent technology related with the communication and the processor. 10. During these features it will be raise the effectiveness, accuracy and transparency.

5.4 LIMITATIONS:

1. The size of the database increases day -by- day increasing the load on the database and the data maintenance activity. 2. Training for simple computer operations is essential for the users working on the system.

Chapter - 6

6.1 Conclusion:

Ø It is fast, reliable and effectiveness and also it voids data redundancy and inconsistency. Ø Number of personnel required is considerably less. Ø This project offers the users to enter the data through the simple and interactive forms. It is very helpful to the client to enter the desired information through it can be most simplicity. Ø The user is mainly concerned about the valid data, what he is entering so there are many checks in every stage of new creation, data entry or updating so the user must enter the exact information that he cannot enter the invalid data or information which can create problems or trouble in the future. Ø The data that can be entered by the user earlier the data and also there is a restriction that he cannot be change in the primary field information which he entered first. So that we can keep the backup for more time. Ø Data storage and recovery become faster and easier to maintain the data for the reason that the data is stored systematically in a single database. Ø It is grateful to me to work on this exciting and challenging project. This project taught me what i have provided practical knowledge of not only the programming in ASP.NET and C#.NET web based application and also some areas Windows Application and SQL Server and also about all handling procedure related with “E-Health care management system”. And It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great in future. This will provide better opportunities to me in future for developing projects independently.

6.2 Future Enhancements:

We can enhance this system by including more facilities like billing system, online appointments, mobile SMS confirmation and alerts, forum and testimonial. Such features enable the users to include more comments into the system.

6.3 References:

1. Esben Dalsgaard, Kare Kjelstorm, Jan Riis (2008) - A Federation Of Web Services For Danish Health Care,ACM, https://portal.acm.org/citation.cfm?id=1373305 Alexander Hoerbst, Elske Ammenwerth (2009) - A Structural Model For Quality Requirements Regarding Electronic Health Records-State Of The Art and First Concepts, IEEE Computer Society, https://portal.acm.org/citation.cfm?id=1556955 3. Norman Sondheimer,Ethan Katsh,Lori Clarke,Leon Osterweil,Daniel Rainey (2009) - Dispute prevention and dispute resolution in networked health information technology,Digital Government Society Of North America, https://portal.acm.org/citation.cfm?id=1556219 4. Michael Borovicka (2008) - DMIS: Design and Prototype Of a Future Clinical eHealth System,ACM, https://portal.acm.org/citation.cfm?id=1497288 5. Bhuvanneswari

Arunachalam,Janet Light (2007) - Middleware architecture for patient care data transmission using wireless networks,ACM, https://portal.acm.org/citation.cfm?id=1280979 Maria Jansson,Christina Mortberg,Anita Mirijamdotter (2008) - Participation in e-home healthcare @ North Calotte,ACM, https://portal.acm.org/citation.cfm?id=1463181 7. Oystein Nytro,Inger Dybdahl Sorby,Peter Karpati (2009) - Query-based requirements engineering for health care information systems:Examples and prospects,IEEE Computer Society, https://portal.acm.org/citation.cfm?id=1556958 8. Erdogan Dogdu (2009) - Semantic web in eHealth,ACM, https://portal.acm.org/citation.cfm?id=1566542 9. Toni Ruohonen,Pekka Neittaanmaki,Jorma Teittinen (2006) - Simulation model for improving the operation of the emergency department of special health care,Winter Simulation Conference, https://portal.acm.org/citation.cfm?id=1218198 10. Santosh Kulkarni,Prathima Agrawal (2008) - Smartphone driven healthcare system for rural communities in developing countries,ACM, https://portal.acm.org/citation.cfm?id=1515758 11. John Fulcher (2003) - The use of smart devices in eHealth,Trinity College Dublin, https://portal.acm.org/citation.cfm?id=963606 12. Bruno Von Niman,Alejandro Rodriguez-Ascaso,Steve Brown,Torbjorn Sund 13. (2007) - User experience design guidelines for telecare (e-health) services,ACM, https://portal.acm.org/citation.cfm?id=1288537 Henry F. Korth and Abraham Silberschatz (1987) - Database System Concepts,ACM, https://portal.acm.org/citation.cfm?id=30075.1096828 Roger. S. Pressman (2005) - Software Engineering A Practitioners Approach,ACM, https://highered.mcgraw-hill.com/sites/0072853182/information_center_view0/ 16. Chris Hart, John Kauffman, Dave Sussman, and Chris Ullman (2006) - beginning C#.Net 2.0. 17 .Lee Purcell, Mary Jane Mara (1999) - The ABCs of Java Script,Sybex 18 Bill Evjen (2006) - Professional ASP.NET 2.0. · FOR DEPLOYMENT AND PACKING ON SERVER www.developer.com www.15seconds.com · FOR SQL www.msdn.microsoft.com www.msdn.microsoft.com · FOR ASP.NET www.asp.net www.asptoday.com www.aspfree.com · UI DESIGN www.glassbox-js.com (template design) www.jquery.com (template design) www.w3schools.com (CSS) www.regular-expressions.info/javascriptexample.html (validations)