A PROJECT REPORT ON

COMPUTER INSTUTUTE MANAGEMENT SYSTEM

Submitted To

UNIVERSITY OF MUMBAI

IN

PARTIAL FULFILMENT OF

T.Y.B.Sc. IN INFORMATION TECHNOLOGY

**Semester VI**

By

**Mr. ---**

Mr. --

**Year 2012-2013**

Project Guide

Prof. ---

Department of Information Technology

ICLES’ M.J. College of Arts, Commerce & Science

**Vashi, Navi Mumbai**

**certificate**

This is to certify that the project entitled “Online Tours And Travels Management System” is successfully completed by MS.ASHWINI BHOR as per the syllabus and in partial fulfillment for the completion of BSc. degree in INFORMATION TECHNOLOGY of University of Mumbai, it is also to certify that this is the original work of the candidate done during the academic year 2012 – 2013.

Exam seat no:

Place:

Date:

Signature of : Signature of :

Guide Principal /In-charg/Co- ordinator

Signature of : Seal :

External Examiner

**ACKNOWLEDGEMENT**

Any successful project is the result of culmination of efforts of more than one individual. Our project is no exception. We have been extremely fortunate in having the help of many people during the course of this project. We would like to thank them all. We express our sincere gratitude to our Internal Guide **Prof. Ameya Shirpurkar** (Professor, Department of Information Technology) for all the encouragement and analytical views on our topic of interest. His constructive suggestions and insight helped us in exploring our project front all possible perspective.

We also thank **Prof. Mrs. Sushama Verulkar** (Head of Department, Information Technology) for his encouragement, support and creative vision. We would also like to thank all the faculty members and our friends for their continuous support throughout these four years. You all were always there to encourage us, ready to help us out in case of any problem. You all will always remain close to our hearts for all your love and support.

We are grateful to the college for giving us an opportunity to perform this project, as a part of fulfillment for Bsc. in Information Technology.

We would also like to thank the Principal of our college **I/C Dr. Jyoti Marwah**, for arranging the project for us, as well as for providing facilities for implementation of the same.

**--**

--

**--**

|  |  |
| --- | --- |
| Index | Page No. |
| **Acknowledgement** | 3 |
| **I. Preliminary Investigation** | 5 |
| (i) Organizational Overview | 6 |
| (ii) Limitations of present system | 7 |
| (v) Feasibility Study | 8 |
| (vi) Stakeholders | 10 |
| (vi) Gantt Chart | 11 |
| **II. System Analysis** | 12 |
| (i) Fact Finding Techniques (Questionnaire, Sample Reports, Forms...) | 14 |
| (ii)System Diagrams | 15-18 |
| (iii)Validation | 19 |
| (iv)Testing Of system | 20 |
| **III. System Design** | 22 |
| (i)Data Structures | 22 |
| **IV. System Coding** |  |
| (i) Screen Layouts & Report Layouts | 24-33 |
| (ii)System Code | 25-64 |
| **V. System Implementation / Uploading** | 65 |
| **VI. Future Enhancements** | 67 |
| **VII. References and Bibliography** | 68 |

**Preliminary Investigation**

**Organizational Overview**

The proposed system deals with all included activities handled by a computer center like enquiry, registration, faculty details, courses, fee information etc. This system is developed to eliminate all the complexities that can even occurred during the handling of a computer center. Now these days often seen that manual handling of a computer institute system has become more tedious task because a lot of many student comes for enquiring about different courses, few of them registered. A lot of courses are being conducted by the computer institute and to maintain faculty detail manually is also a tuff job. Since these works does manually thus it takes more time and there are chances of errors such as wrong fee details, wrong student registration etc.

**LIMITATIONS OF THE EXISTING SYSTEM**

The proposed system deals with all included activities handled by a computer center like enquiry, registration, faculty details, courses, fee information etc. This system is developed to eliminate all the complexities that can even occurred during the handling of a computer center. Now these days often seen that manual handling of a computer institute system has become more tedious task because a lot of many student comes for enquiring about different courses, few of them registered. A lot of courses are being conducted by the computer institute and to maintain faculty detail manually is also a tuff job. Since these works does manually thus it takes more time and there are chances of errors such as wrong fee details, wrong student registration etc. So to reduce these faults Computer Institute Management System has been developed. The proposed system. Computer Institute Management System deals with all the activities done by any computer institute. So this project is developed for all those institutes who are dealing with enquiry, registration, course details, and fee details processes. This system is more flexible and efficient than manual work and reduce the total time consume at error chances. As we know that if we use a new system on existing system then we must want to know about objective of proposed system. Following are the some key objective of proposed system that makes it better than existing system.

**FEASIBILITY STUDY**

It is used to define a precise cost & benefits of the software system. The contents of feasibility study phase are highly dependent upon the type of software developer & application at hand.

The purpose of this phase is to feasibility study document that evaluates the cost & benefits of the proposed application.

The feasibility study is a sort of simulation of a future development process through which it is possible to derive the information that helps decide whether the development is worth while and, if so which development process should be followed.

In sum, the feasibility study tries to anticipate future scenarios of software development. Its result is a document that should contain at least the following topics:

1. A definition of the problem

2. Alternative solution & their expected benefits.

3. Required resources, cost & delivery date in each proposed alternative solution.

**TECHNICAL FEASIBILITY**:

Considering following aspects feasibility study of the system is carried out. The feasibility study is deciding factor which determines what the technical resources are essential for designing & developing of system.

In order top know the user use regarding technical resources below are the resources required for system development.

**ECONOMICAL FEASIBILITY:**

Economical feasibility is the study of actual cost & benefits of system.

The actual cost of the system is calculated in the economical study so that the user or the management come to know the actual cost of implementation of this project & the benefits gained from this project as the output.

The exact period when the project will prove to be a beneficial is taken in to consideration while designing the system.

The required cost of my proposed system is low because all hardware & software resources are already available. We are not using any new technologies so we don’t require hiring the people from outside.

Hence we concluded that my system is economically feasible.

**OPERATIONAL FEASIBILITY:**

Total working capacity will be improved.

Because of implementation of computerized system time is nicely organized .

A manpower required for clerical work will be reduced.

The system is operationally feasible because of above tested benefits.

System is operationally feasible because of the above tests.

**STAKEHOLDERS**

Stakeholders are anyone who has an interest in the project. Project stakeholders are individuals and organizations that are actively involved in the project, or whose interests may be affected as a result of project execution or project completion. They may also exert influence over the project’s objectives and outcomes. The project management team must identify the stakeholders, determine their requirements and expectations, and, to the extent possible, manage their influence in relation to the requirements to ensure a successful project.

**The following are examples of project stakeholders:**

1. **Project Developers**

Project Developers is one who develops software for customer. In my project I am the first stakeholder i.e. Project Developer.

1. **Project customer**

Project Customer is one who pays for the developed software. In my project the Chairman of Technovision Automation Pvt. Ltd. is the second stakeholder i.e. Project Customer.

1. **Project Testers**

Project Testers are those who test the software. In my project myself, my project guide and Technical Department of TAPL are the fourth stakeholders i.e. Project Testers.

**GANTT CHART: INDICATIONS OF THE GANTT CHART: Time Given**

**Time Taken**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SR NO.** | **ACTIVITIES** | **OCT**  **01-6-2012**  **To**  **30-6-2012** | **NOV**  **01-7-2012**  **To**  **30-7-2012** | **DEC**  **01-8-2012**  **To**  **30-8-2012** | **JAN**  **01-9-2012**  **To**  **30-9-2012** | **FEB**  **01-10-2012**  **To**  **30-10-2012** | **MAR**  **01-11-2012**  **To**  **30-11-2012** | **APR**  **01-12-2012**  **To**  **30-12-2012** |
| 1. | **Planning phase:**   * **Define the problem** * **Produce project schedule** * **Launch the project** |  |  |  |  |  |  |  |
| 2. | **Analysis phase:**   * **Gather information** * **Define the system** |  |  |  |  |  |  |  |
| 3. | **Design phase:**   * **Complete application design** * **Design user interface** * **Design and integrate database and system control** |  |  |  |  |  |  |  |
| 4. | **Coding phase:**   * **Writing code for module** * **Integrating module** |  |  |  |  |  |  |  |
| 5. | **Testing phase:**   * **Unit testing** * **System testing** |  |  |  |  |  |  |  |
| 6. | **Implementation phase:** |  |  |  |  |  |  |  |

SYSTEM ANALYSIS

**3.1 REQUIREMENT ANALYSIS**

**3.1.1 FUNCTIONAL REQUIREMENTS**

**a) Strong Data Validation:**

There is possibility that user might enter wrong data and wrong data may cause inconsistency to the database and hence to the system. To avoid this, data should be validated whenever entered.

**b) Automatic updating of the database:**

After any transaction is performed, it is necessary that the updating should be reflected in the database without any inconsistency.

**c) Provide efficiency querying based on user requests:**

The major purpose is to generate efficient reports on any user request. This will be done by our query processing system, which should be able to process any combination of queries will be done dynamically at run time depending on the user

**3.1.2 EXTERNAL INTERFACE REQUIREMENTS**

**a) User friendly interface:**

The interface should be developed in such a manner that it is very user friendly, this not only improve interaction but also saves data entry time.

**b) Making well designed forms for capturing data:**

The forms for capturing the data should be well-designed using pop-down menus and drag & drop facilities, which reduce the data entry effort on the part of the user.

**3.1.3 PERFORMANCE REQUIREMENTS**

**a) Security:**

All users are not allowed to access the database. Hence there is a need to check authority of every user. Username and Password validation helps to deny unauthorized access to the system.

There are 2 main types of users who will be using the software

They are:-

1) Admin

2) User

Each user is given the specific rights to access the data in Read only, Read Write, Delete.

**Fact Finding techniques**

It is used to collect the data required in the development of the Project.

**Internet:**

Product information, Description and picture are collected from all over the internet the data is collected from various websites and articles. These data are then sorted and are added to the database

**3.2 ERD**

Manages

Maintains

tbl\_Student

Register

Student

tbl\_Staff

Register

Computer Institute Management System

Maintains

tbl\_Course

tbl\_Login

tbl\_Fee

* 1. **DATA FLOW DIAGRAM**

.

Search and delete the Data

Admin

Check User Data

Details

Add Staff Details

Add Course Details

User

Add Student Details

Enter User Detail

User User Profile

Details

Student Student Record

Details

Course Course Details

Details

Staff Staff Record

Details

* 1. **CONTEXT LEVEL DAIGRAM**

Student

4. Course Details

Admin

8.Fees Details

7.Fees Details

5.Staff Details

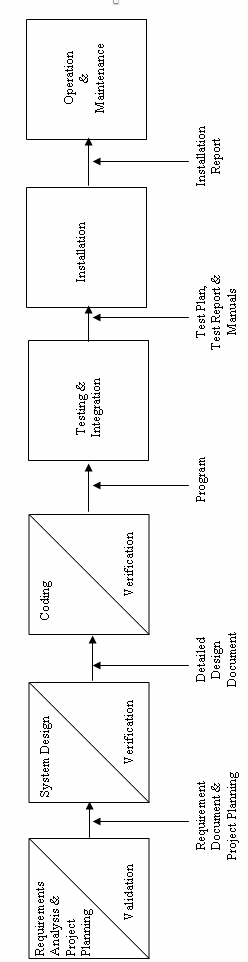
3.Course Details

2. Student Details

6.Staff Details

1.Login

Staff



**Validation:**

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data whenever an invalid data in keyed in the system immediately prompts the user.

Form Description

Login user and password should not be empty or invalid.

Registration All the user details must be correct.

### Testing of System

Testing is asset of activities that can be planned in advanced and conducted systematically.

For this reason a template for software testing a set into which we can specific test case design techniques and testing methods should be defined for the software process.

A strategy for software testing must accommodate low-level test that are necessary to verify that a small source code segment can be correctly implemented as well as high –level tests that validate major system functions against customer requirements.

**TYPES OF TESTING:**

RMS went through the following types of testing: -

1. **Alpha Testing: -**

Testing after code is mostly complete or contains most of the functional and prior to end user being involved. Sometimes selected group of users as involved. More often this testing will be performed in house or by an outside testing firm in close cooperation with the software engineering department

1. **Beta Testing: -**

Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.

1. **Functional Testing: -**

Testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the modules performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed do.

1. **Configuration Testing: -**

Testing to determine how well the product works with a broad of the hardware/peripheral equipment configurations as on the different operating systems and software.

1. **Pilot Testing: -**

Testing that involves the users just before actual release to ensure that users become familiar with the release contents and ultimately accept it. Often is considered a Move-to-Production activity for ERP release or a beta test for commercial products. Typically involves many users, is conducted over a short period of time and is tightly controlled. (See beta testing).

1. **System Integration Testing: -**

Testing a specific hardware/software installation. This is typically performed on a COTS (commercial off the shelf )system or any other system comprised or the disparate parts where custom configurations and /or unique installation are the norm .

1. **Software Testing: -**

The process of exercising software is with the intent of ensuring that the software system meets its requirements and the user expectations and doesn’t file in an unacceptable manner . The organization and the management of the individuals or groups dong this work is not relevant. This term is often applied to commercial products such as Internet application. (Contrast with independent verification and validation)

1. **Security testing: -**

Testing of database and network software in order to keep company data and resources from mistaken/ accidental users, hackers and other malevolent attackers

1. **Installation Testing: -**

Testing with the intent of determining if the product will install on a variety of platforms and how easily it installs

10. **Compatibility Testing: -**

Testing used to determine whether other system software components such as browsers, utilities and competing software would conflict with the software being tested.

**SYSTEM DESIGN**

# DATA STRUCTURE

**1) Table Name: tbl\_Course**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| COURSE\_No | 50 | varchar(50) | Course Number |
| COURSE\_COURSENAME | 50 | varchar(50) | Course Name |
| COURSE\_DURATION | 50 | varchar(50) | Duration |
| COURSE\_FEES | 50 | varchar(50) | Fess |
| COURSE\_BATCHTIMING | 50 | varchar(50) | Batch Timing |
| COURSE\_FACULTY | 50 | varchar(50) | Faculty |

**2) Table Name: tbl\_Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| username | 50 | varchar(50) | Username |
| password | 50 | varchar(50) | Password |

**3) Table Name: tbl\_Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Staff\_N0 | 50 | varchar(50) | ID |
| Staff\_NAME | 50 | varchar(50) | Name |
| Staff\_DOB |  | smalldatetime | Date of Birth |
| Staff\_QUALIFICATION | 50 | varchar(50) | Qualification |
| Staff\_EXPERIENCE | 50 | varchar(50) | Experience |
| Staff\_EMAILID | 50 | varchar(50) | Email ID |
| Staff\_PHONE | 50 | varchar(50) | Contact Number |
| Staff\_ADDRESS | 50 | varchar(50) | Address |

**4) Table Name: tbl\_Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Student\_ROLLNO | 50 | varchar(50) | Roll Number |
| Student\_NAME | 50 | varchar(50) | Name |
| Student\_ADDRESS | 50 | varchar(50) | Address |
| Student\_PHONE | 50 | varchar(50) | Contact Number |
| Student\_QUALIFICATION | 50 | varchar(50) | Qualification |
| Student\_GENDER | 50 | varchar(50) | Gender |
| Student\_DOB |  | smalldatetime | Date of Birth |
| Student\_EMAIL | 50 | varchar(50) | Email ID |
| Student\_OCCUPATION | 50 | varchar(50) | Occupation |
| Student\_KNOWLEDGE | 50 | varchar(50) | Konowledge |
| Student\_TOTALFEES | 50 | varchar(50) | Total fees |

**5) Table Name: tbl\_Fee**

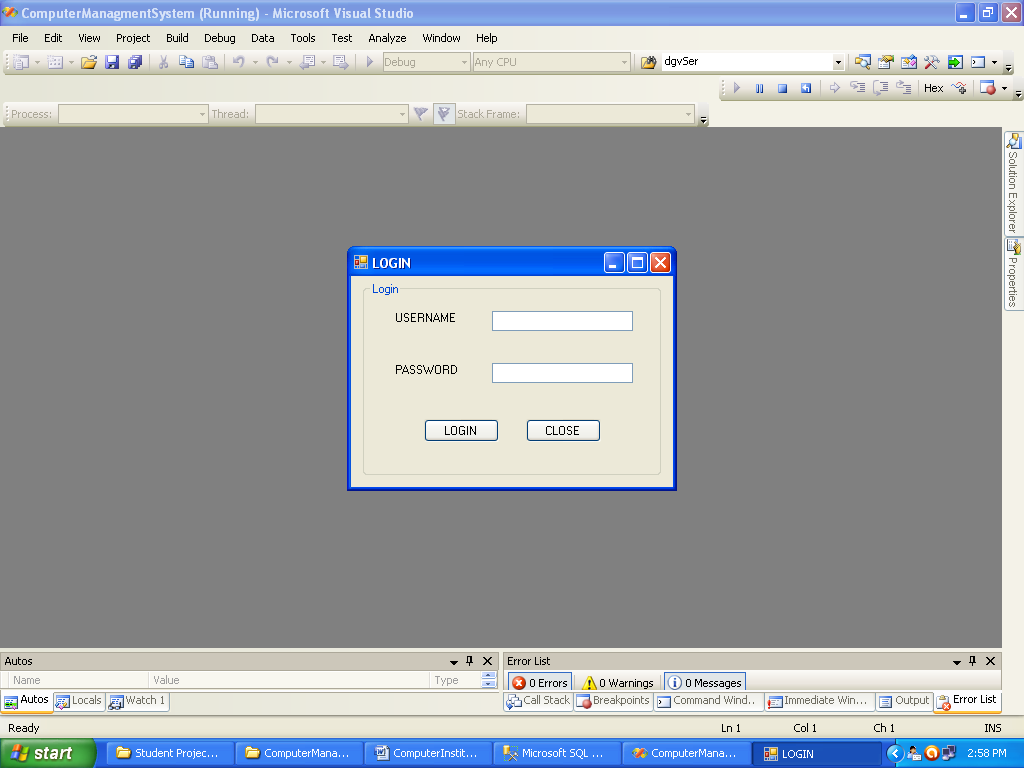
|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Fee\_ROLLNO | 50 | varchar(50) | Roll Number |
| Fee\_RECEIPTNO | 50 | varchar(50) | Receipt Number |
| Fee\_Date |  | smalldatetime | Date |
| Fee\_AMOUNT | 50 | varchar(50) | Amount |
| Fee\_ModeOfPayment | 50 | varchar(50) | Mode of Payment |
| Fee\_BANKNAME | 50 | varchar(50) | Bank Name |
| Fee\_CHEQUENO | 50 | varchar(50) | Cheque Number |
| Fee\_BAMOUNT | 50 | varchar(50) | Balance Amount |
| Fee\_BasicFees | 50 | varchar(50) | Basic Fees |
| Fee\_DiscountPercentage | 50 | varchar(50) | Discount Percentage |
| Fee\_DiscountAmount | 50 | varchar(50) | Discount Amount |
| Fee\_TotalFees | 50 | varchar(50) | Total Fees |

**6) Table Name: Purchases**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Serial\_No | 50 | varchar(50) | Serial No |
| Name | 50 | varchar(50) | Item name |
| Amount | 50 | varchar(50) | Amount |

**SCREEN LAYOUTS AND FORM DESIGNING*.***

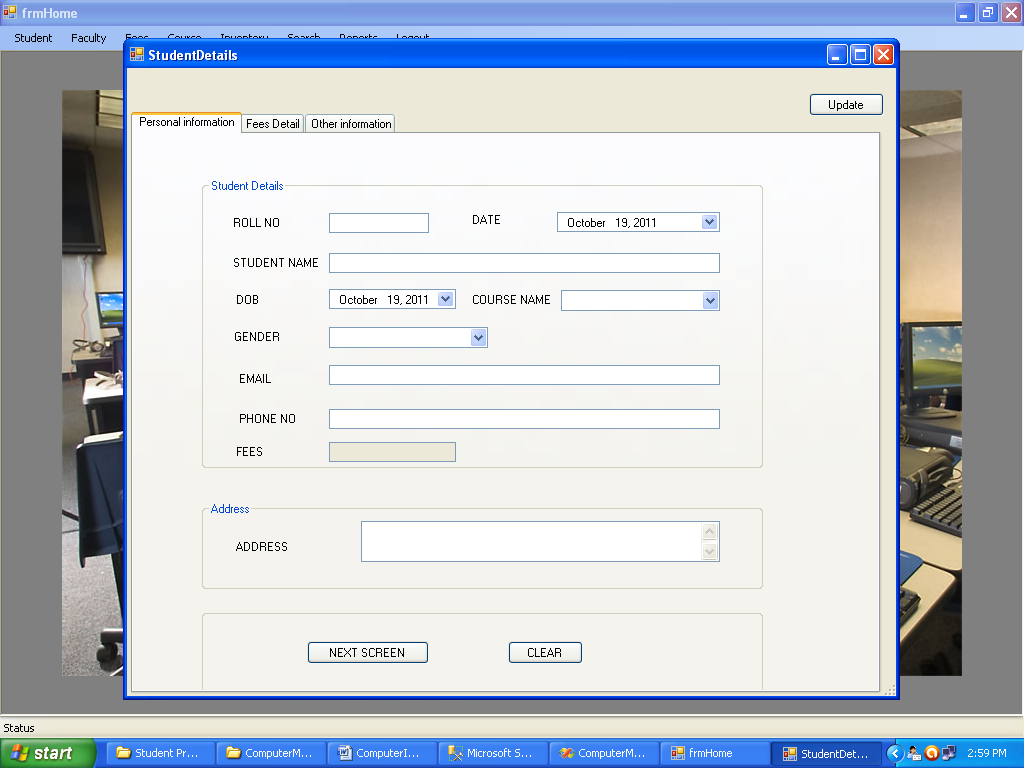
**1) Login**



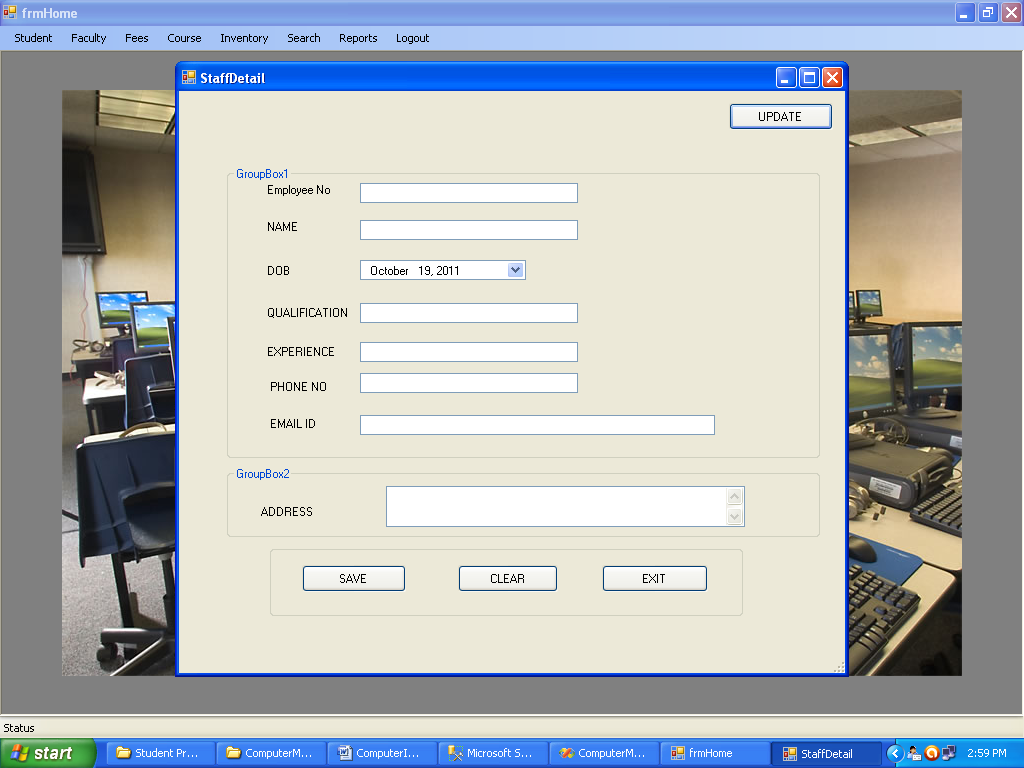
**2) Home**

****

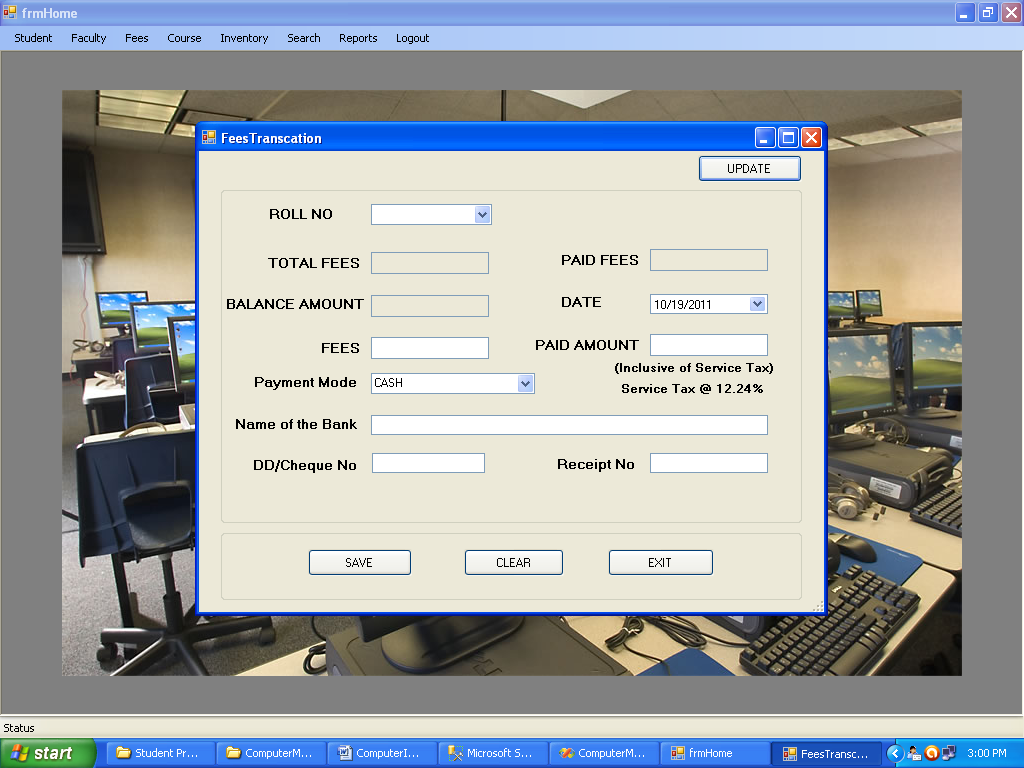
**3) Student**

****

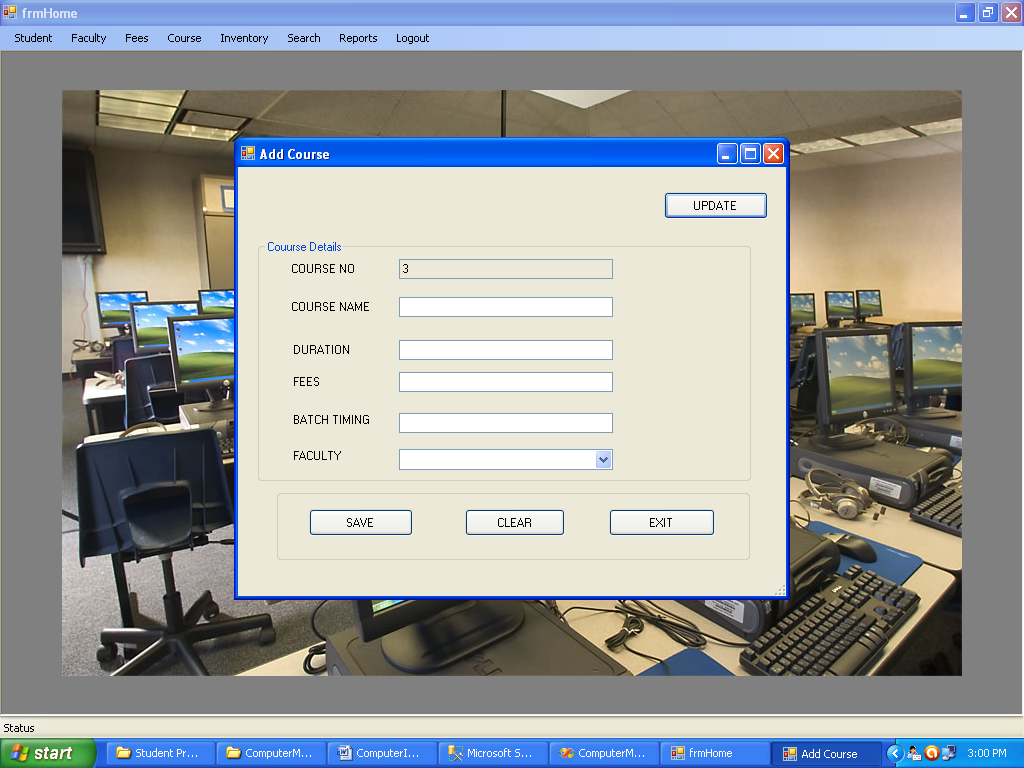
**4) Faculty**

****

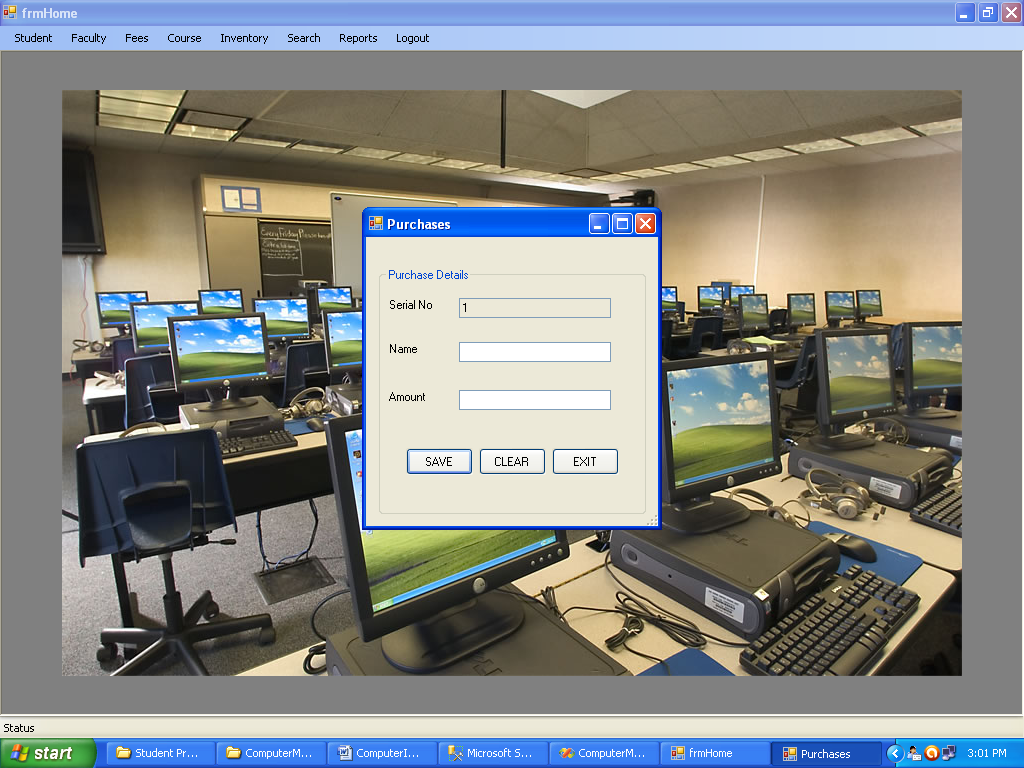
**5) Fees**

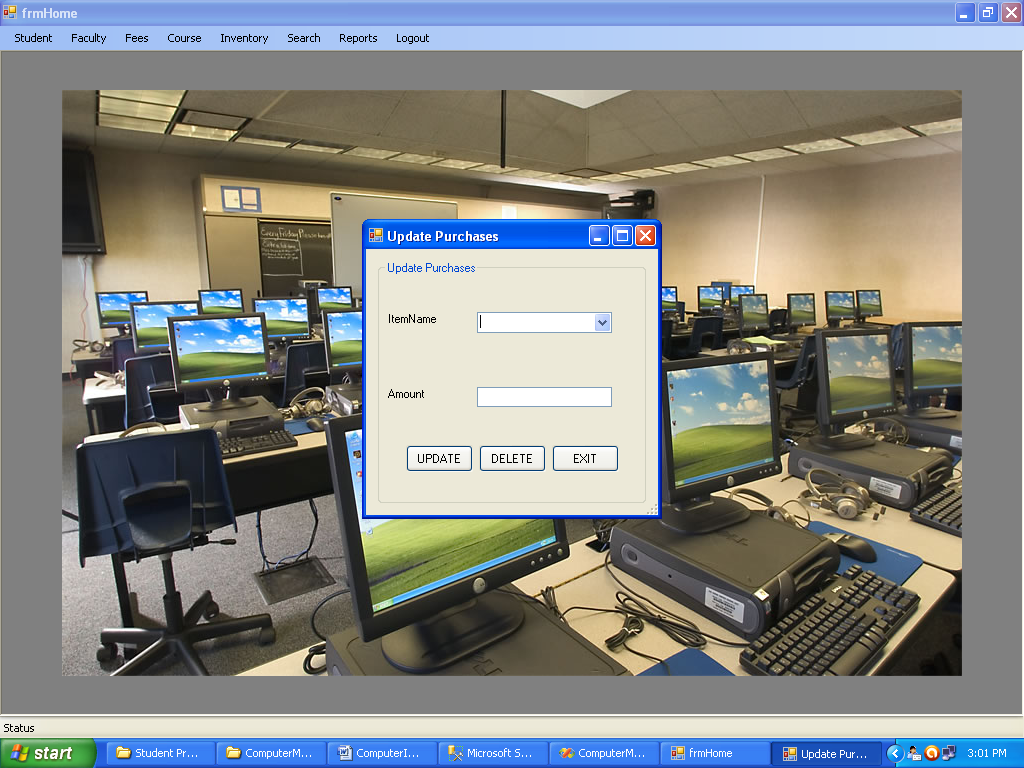
****

**6) Course**

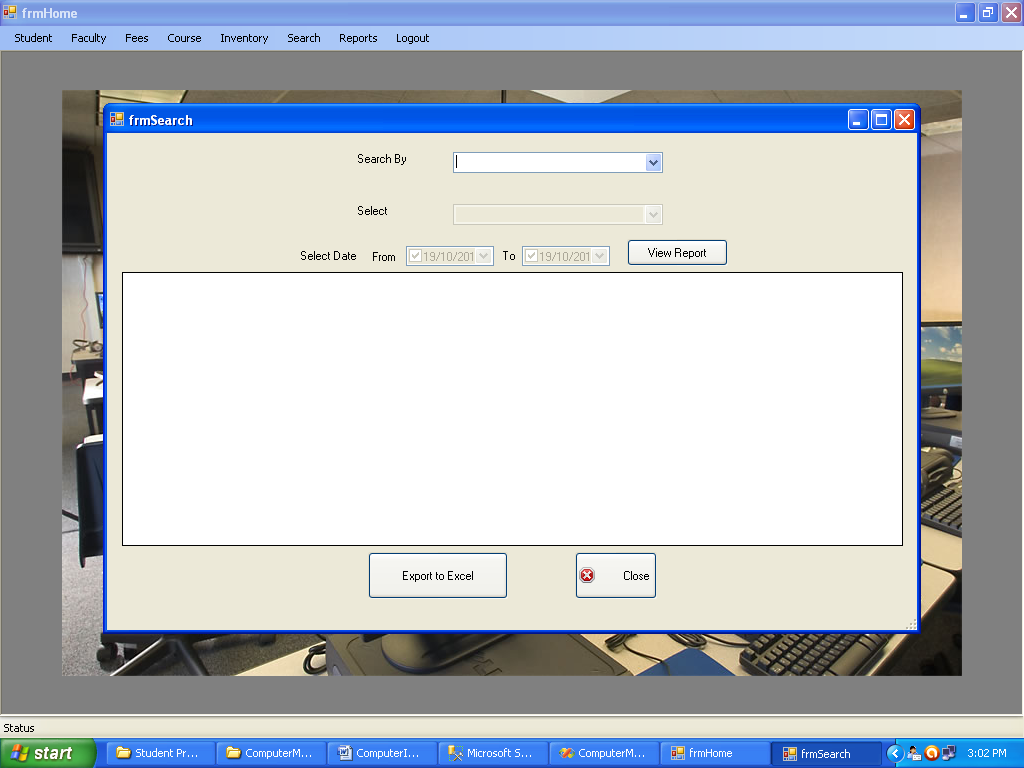
****

**7) Inventory**

****

****

**8) Search**

****

**9) Reports**

****

**System Code**

**Login Page:-**

using System.Diagnostics;

using System;

using System.Xml.Linq;

using System.Windows.Forms;

using System.Collections;

using System.Drawing;

using Microsoft.VisualBasic;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Data.SqlClient;

namespace ComputerManagmentSystem

{

public partial class FrmLogin

{

public FrmLogin()

{

InitializeComponent();

//Added to support default instance behavour in C#

if (defaultInstance == null)

defaultInstance = this;

}

#region Default Instance

private static FrmLogin defaultInstance;

/// <summary>

/// Added by the VB.Net to C# Converter to support default instance behavour in C#

/// </summary>

public static FrmLogin Default

{

get

{

if (defaultInstance == null)

{

defaultInstance = new FrmLogin();

defaultInstance.FormClosed += new FormClosedEventHandler(defaultInstance\_FormClosed);

}

return defaultInstance;

}

}

static void defaultInstance\_FormClosed(object sender, FormClosedEventArgs e)

{

defaultInstance = null;

}

#endregion

SqlConnection con = new SqlConnection("Data Source=.\\sqlexpress;Initial Catalog=D:\\READY PROJECTS\\COMPUTERMANAGMENTSYSTEM\\DB\_CIMS.MDF;Integrated Security=True");

public void FrmLogin\_Load(System.Object sender, System.EventArgs e)

{

}

public void btnLogin\_Click(System.Object sender, System.EventArgs e)

{

string a;

string b;

a = txtUsername.Text;

b = txtPassword.Text;

int flag = 0;

con.Open();

SqlCommand cmd = new SqlCommand("select \* from tbl\_Login", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

if (a == rd[0].ToString().Trim() && b == rd[1].ToString().Trim())

{

flag = 1;

break;

}

else

{

flag = 0;

}

}

if (flag == 1)

{

//MsgBox("Login Successfull", MsgBoxStyle.OkOnly, "Done")

this.Hide();

frmHome.Default.Show();

}

else

{

Interaction.MsgBox("User Name or Password may be wrong.", MsgBoxStyle.Critical, "Error");

}

con.Close();

}

public void btnClose\_Click(System.Object sender, System.EventArgs e)

{

this.Close();

}

public void txtPassword\_KeyDown(System.Object sender, System.Windows.Forms.KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

btnLogin\_Click(sender, e);

}

}

}

}

**Home:-**

using System.Diagnostics;

using System;

using System.Xml.Linq;

using System.Windows.Forms;

using System.Collections;

using System.Drawing;

using Microsoft.VisualBasic;

using System.Data;

using System.Collections.Generic;

using System.Linq;

namespace ComputerManagmentSystem

{

public partial class frmHome

{

public frmHome()

{

InitializeComponent();

//Added to support default instance behavour in C#

if (defaultInstance == null)

defaultInstance = this;

}

#region Default Instance

private static frmHome defaultInstance;

/// <summary>

/// Added by the VB.Net to C# Converter to support default instance behavour in C#

/// </summary>

public static frmHome Default

{

get

{

if (defaultInstance == null)

{

defaultInstance = new frmHome();

defaultInstance.FormClosed += new FormClosedEventHandler(defaultInstance\_FormClosed);

}

return defaultInstance;

}

}

static void defaultInstance\_FormClosed(object sender, FormClosedEventArgs e)

{

defaultInstance = null;

}

#endregion

private void ShowNewForm(object sender, EventArgs e)

{

// Create a new instance of the child form.

System.Windows.Forms.Form ChildForm = new System.Windows.Forms.Form();

// Make it a child of this MDI form before showing it.

ChildForm.MdiParent = this;

m\_ChildFormNumber++;

ChildForm.Text = (string) ("Window " + m\_ChildFormNumber);

ChildForm.Show();

}

private void OpenFile(object sender, EventArgs e)

{

OpenFileDialog OpenFileDialog = new OpenFileDialog();

OpenFileDialog.InitialDirectory = (new Microsoft.VisualBasic.Devices.ServerComputer()).FileSystem.SpecialDirectories.MyDocuments;

OpenFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*";

if (OpenFileDialog.ShowDialog(this) == System.Windows.Forms.DialogResult.OK)

{

string FileName = OpenFileDialog.FileName;

// TODO: Add code here to open the file.

}

}

private void SaveAsToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog SaveFileDialog = new SaveFileDialog();

SaveFileDialog.InitialDirectory = (new Microsoft.VisualBasic.Devices.ServerComputer()).FileSystem.SpecialDirectories.MyDocuments;

SaveFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*";

if (SaveFileDialog.ShowDialog(this) == System.Windows.Forms.DialogResult.OK)

{

string FileName = SaveFileDialog.FileName;

// TODO: Add code here to save the current contents of the form to a file.

}

}

private void ExitToolsStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void CutToolStripMenuItem\_Click(object sender, EventArgs e)

{

// Use My.Computer.Clipboard to insert the selected text or images into the clipboard

}

private void CopyToolStripMenuItem\_Click(object sender, EventArgs e)

{

// Use My.Computer.Clipboard to insert the selected text or images into the clipboard

}

private void PasteToolStripMenuItem\_Click(object sender, EventArgs e)

{

//Use My.Computer.Clipboard.GetText() or My.Computer.Clipboard.GetData to retrieve information from the clipboard.

}

private void CascadeToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.Cascade);

}

private void TileVerticalToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.TileVertical);

}

private void TileHorizontalToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.TileHorizontal);

}

private void ArrangeIconsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.ArrangeIcons);

}

private void CloseAllToolStripMenuItem\_Click(object sender, EventArgs e)

{

// Close all chi ld forms of the parent.

foreach (Form ChildForm in this.MdiChildren)

{

ChildForm.Close();

}

}

private int m\_ChildFormNumber;

public void frmHome\_Load(System.Object sender, System.EventArgs e)

{

}

public void AddStudentToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

}

public void AddFacultyToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

}

public void FeesToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

}

public void AddCourseToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

}

public void ReportsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmSearch f = new frmSearch();

f.ShowDialog();

}

public void FessDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmViewReport obj = new frmViewReport();

obj.View\_Report("select \* from tbl\_Fee", "Fees");

obj.ShowDialog();

}

public void StudentDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmViewReport obj = new frmViewReport();

obj.View\_Report("select \* from tbl\_Student", "Student");

obj.ShowDialog();

}

public void FacultyDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmViewReport obj = new frmViewReport();

obj.View\_Report("select \* from tbl\_Staff", "Faculty");

obj.ShowDialog();

}

public void CourseDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmViewReport obj = new frmViewReport();

obj.View\_Report("select \* from tbl\_Course", "Course");

obj.ShowDialog();

}

public void ToolStripMenuItem1\_Click(System.Object sender, System.EventArgs e)

{

}

public void PurchasesToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmPurchases f = new frmPurchases();

f.Show();

}

public void UpdatePurchasesToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmUpdatePurchases f = new frmUpdatePurchases();

f.ShowDialog();

}

public void AddCourseToolStripMenuItem1\_Click(System.Object sender, System.EventArgs e)

{

addCourse f = new addCourse();

f.ShowDialog();

}

public void AddFeesDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

FeesTranscation f = new FeesTranscation();

f.ShowDialog();

}

public void AddStudentToolStripMenuItem1\_Click(System.Object sender, System.EventArgs e)

{

StudentDetails f = new StudentDetails();

f.ShowDialog();

}

public void AddFacultyToolStripMenuItem1\_Click(System.Object sender, System.EventArgs e)

{

StaffDetail f = new StaffDetail();

f.ShowDialog();

}

public void PurchaseDetailsToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

frmViewReport obj = new frmViewReport();

obj.View\_Report("select \* from tbl\_Purchases", "Purchase");

obj.ShowDialog();

}

public void LogoutToolStripMenuItem\_Click(System.Object sender, System.EventArgs e)

{

this.Close();

}

}

}

**Student:-**

using System.Diagnostics;

using System;

using System.Xml.Linq;

using System.Windows.Forms;

using System.Collections;

using System.Drawing;

using Microsoft.VisualBasic;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Data.SqlClient;

namespace ComputerManagmentSystem

{

public partial class StudentDetails

{

public StudentDetails()

{

InitializeComponent();

//Added to support default instance behavour in C#

if (defaultInstance == null)

defaultInstance = this;

}

#region Default Instance

private static StudentDetails defaultInstance;

/// <summary>

/// Added by the VB.Net to C# Converter to support default instance behavour in C#

/// </summary>

public static StudentDetails Default

{

get

{

if (defaultInstance == null)

{

defaultInstance = new StudentDetails();

defaultInstance.FormClosed += new FormClosedEventHandler(defaultInstance\_FormClosed);

}

return defaultInstance;

}

}

static void defaultInstance\_FormClosed(object sender, FormClosedEventArgs e)

{

defaultInstance = null;

}

#endregion

SqlConnection con = new SqlConnection("Data Source=.\\sqlexpress;Initial Catalog=D:\\READY PROJECTS\\COMPUTERMANAGMENTSYSTEM\\DB\_CIMS.MDF;Integrated Security=True");

public void GroupBox1\_Enter(System.Object sender, System.EventArgs e)

{

}

public void Label3\_Click(System.Object sender, System.EventArgs e)

{

}

public void GroupBox2\_Enter(System.Object sender, System.EventArgs e)

{

}

public void clear()

{

txtRollNo.Clear();

dtpDate.ResetText();

txtStudentName.Clear();

dtpDob.ResetText();

cboCourseName.ResetText();

cboGender.ResetText();

txtEmail.Clear();

txtPhoneNo.Clear();

txtTotalfee.Clear();

txtAddress.Clear();

txtFee.Clear();

txtDicountAmount.Clear();

txtDiscountPercentage.Clear();

txtTotalfee.Clear();

txtAmountPaid.Clear();

txtReceiptNo.Clear();

cboPaymentMode.ResetText();

txtNameOfBank.Clear();

txtChequeNo.Clear();

txtFatherName.Clear();

txtMotherName.Clear();

txtHusbandName.Clear();

cboQualification.ResetText();

cboOccupation.ResetText();

txtKnowledge.Clear();

}

public void loadStudent()

{

con.Open();

SqlCommand cmd = new SqlCommand("select Student\_NAME from tbl\_Student", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

cboStudent.Items.Add(rd[0]);

}

con.Close();

}

public void StudentDetails\_Load(System.Object sender, System.EventArgs e)

{

txtNameOfBank.Enabled = false;

txtChequeNo.Enabled = false;

loadStudent();

btnUpdate.Visible = false;

btnDelete.Visible = false;

pnlSearch.Visible = false;

SqlDataAdapter adapter = new SqlDataAdapter("select [Roll Number]=Student\_ROLLNO,[Name]=Student\_NAME,[Address]=Student\_ADDRESS,[Contact Number]=Student\_PHONE from tbl\_Student", con);

DataSet ds = new DataSet();

adapter.Fill(ds);

dgvSearch.DataSource = ds.Tables[0];

con.Open();

SqlCommand cmd = new SqlCommand("select COURSE\_NAME from tbl\_Course", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

cboCourseName.Items.Add(rd[0]);

}

con.Close();

}

public void btnNextScreen\_Click(System.Object sender, System.EventArgs e)

{

TabControl1.SelectedTab = TabPage2;

}

public void btnPreviousScreen\_Click(System.Object sender, System.EventArgs e)

{

TabControl1.SelectedTab = TabPage1;

}

public void btnNextScreen2\_Click(System.Object sender, System.EventArgs e)

{

TabControl1.SelectedTab = TabPage3;

}

public void btnPreviousScreen3\_Click(System.Object sender, System.EventArgs e)

{

TabControl1.SelectedTab = TabPage2;

}

public void btnClear\_Click(System.Object sender, System.EventArgs e)

{

txtRollNo.Clear();

dtpDate.ResetText();

txtStudentName.Clear();

dtpDob.ResetText();

cboCourseName.ResetText();

cboGender.ResetText();

txtEmail.Clear();

txtPhoneNo.Clear();

txtTotalfee.Clear();

txtAddress.Clear();

}

public void btnClear2\_Click(System.Object sender, System.EventArgs e)

{

txtFee.Clear();

txtDicountAmount.Clear();

txtDiscountPercentage.Clear();

txtTotalfee.Clear();

txtAmountPaid.Clear();

txtReceiptNo.Clear();

cboPaymentMode.ResetText();

txtNameOfBank.Clear();

txtChequeNo.Clear();

}

public void BtnClear3\_Click(System.Object sender, System.EventArgs e)

{

txtFatherName.Clear();

txtMotherName.Clear();

txtHusbandName.Clear();

cboQualification.ResetText();

cboOccupation.ResetText();

txtKnowledge.Clear();

}

public void btnSave1\_Click(System.Object sender, System.EventArgs e)

{

if (txtRollNo.Text == "" || cboPaymentMode.Text == "")

{

Interaction.MsgBox("Details are Incomplete", MsgBoxStyle.Exclamation, null);

}

else

{

con.Open();

string s = "Insert into tbl\_Student(Student\_ROLLNO,Student\_NAME,Student\_FNAME,Student\_MNAME,Student\_HNAME,Student\_ADDRESS,Student\_PHONE,Student\_QUALIFICATION,Student\_GENDER,Student\_DOB,Student\_EMAIL,Student\_OCCUPATION,Student\_CNAME,Student\_KNOWLEDGE,Student\_TOTALFEES) values(\'" + txtRollNo.Text + "\',\'" + txtStudentName.Text + "\',\'" + txtFatherName.Text + "\',\'" + txtMotherName.Text + "\',\'" + txtHusbandName.Text + "\',\'" + txtAddress.Text + "\',\'" + txtPhoneNo.Text + "\',\'" + cboQualification.Text + "\',\'" + cboGender.Text + "\',\'" + dtpDob.Value.ToString() + "\',\'" + txtEmail.Text + "\',\'" + cboOccupation.Text + "\',\'" + cboCourseName.Text + "\',\'" + txtKnowledge.Text + "\',\'" + txtAmountPaid.Text + "\')";

SqlCommand cmd1 = new SqlCommand(s, con);

cmd1.ExecuteNonQuery();

con.Close();

con.Open();

string s1 = "Insert into tbl\_Fee(Fee\_ROLLNO,Fee\_RECEIPTNO,Fee\_Date,Fee\_BasicFees,Fee\_DiscountPercentage,Fee\_DiscountAmount,Fee\_TotalFees,Fee\_AMOUNT,Fee\_ModeOfPayment,Fee\_BANKNAME,Fee\_CHEQUENO) values(\'" + txtRollNo.Text + "\',\'" + txtReceiptNo.Text + "\',\'" + dtpDate.Value.ToString() + "\',\'" + txtFee.Text + "\',\'" + txtDiscountPercentage.Text + "\',\'" + txtDicountAmount.Text + "\',\'" + txtTotalfee.Text + "\',\'" + txtAmountPaid.Text + "\',\'" + cboPaymentMode.Text + "\',\'" + txtNameOfBank.Text + "\',\'" + txtChequeNo.Text + "\')";

SqlCommand cmd2 = new SqlCommand(s1, con);

cmd2.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Inserted");

clear();

}

}

public void cboCourseName\_SelectedIndexChanged(System.Object sender, System.EventArgs e)

{

con.Open();

SqlCommand cmd = new SqlCommand("select COURSE\_FEES from tbl\_Course where COURSE\_NAME=\'" + cboCourseName.Text + "\'", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

txtTotalFees.Text = (string) (rd[0].ToString());

txtFee.Text = (string) (rd[0].ToString());

}

con.Close();

}

public void txtDiscountPercentage\_Leave(System.Object sender, System.EventArgs e)

{

decimal fees;

decimal DisPercentage;

decimal DiscAmt;

decimal totFees;

decimal tax;

decimal taxAmt;

decimal netAmt;

tax = 12.04M;

if (txtFee.Text == "" || txtDiscountPercentage.Text == "")

{

}

else

{

fees = decimal.Parse(txtFee.Text);

DisPercentage = decimal.Parse(txtDiscountPercentage.Text);

DiscAmt = fees \* DisPercentage / 100;

txtDicountAmount.Text = DiscAmt.ToString();

totFees = fees - DiscAmt;

txtTotalfee.Text = totFees.ToString();

taxAmt = (decimal) ((double) totFees \* 12.04 / 100);

netAmt = totFees + taxAmt;

txtAmountPaid.Text = netAmt.ToString();

}

//rate = Decimal.Parse(txtRate.Text)

//qty = Decimal.Parse(txtQuantity.Text)

//amt = rate \* qty

//lblAmount.Text = amt.ToString

}

public void BtnUpdateSearch\_Click(System.Object sender, System.EventArgs e)

{

pnlSearch.Visible = true;

btnUpdate.Visible = true;

btnDelete.Visible = true;

btnSave.Visible = false;

BtnClear3.Visible = false;

TabControl1.Visible = false;

}

public void cboStudent\_SelectedIndexChanged(System.Object sender, System.EventArgs e)

{

dgvSearch.Columns.Clear();

con.Open();

SqlDataAdapter adapter = new SqlDataAdapter("select [Roll Number]=Student\_ROLLNO,[Name]=Student\_NAME,[Address]=Student\_ADDRESS,[Contact Number]=Student\_PHONE from tbl\_Student where Student\_NAME=\'" + cboStudent.Text + "\'", con);

DataSet ds = new DataSet();

adapter.Fill(ds);

dgvSearch.DataSource = ds.Tables[0];

con.Close();

}

public void dgvSearch\_CellContentClick(System.Object sender, System.Windows.Forms.DataGridViewCellEventArgs e)

{

}

public void dgvSearch\_CellMouseDoubleClick(System.Object sender, System.Windows.Forms.DataGridViewCellMouseEventArgs e)

{

TabControl1.Visible = true;

con.Open();

string str = dgvSearch.CurrentRow.Cells["Roll Number"].Value.ToString();

SqlCommand cmd = new SqlCommand("select \* from tbl\_Student where Student\_ROLLNO=\'" + str + "\'", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

txtRollNo.Text = (string) (rd["Student\_ROLLNO"].ToString());

txtStudentName.Text = (string) (rd["Student\_NAME"].ToString());

txtFatherName.Text = (string) (rd["Student\_FNAME"].ToString());

txtMotherName.Text = (string) (rd["Student\_MNAME"].ToString());

txtHusbandName.Text = (string) (rd["Student\_HNAME"].ToString());

txtAddress.Text = (string) (rd["Student\_ADDRESS"].ToString());

txtPhoneNo.Text = (string) (rd["Student\_PHONE"].ToString());

cboQualification.SelectedText = (string) (rd["Student\_QUALIFICATION"].ToString());

cboGender.SelectedText = (string) (rd["Student\_GENDER"].ToString());

dtpDob.Text = (string) (rd["Student\_DOB"].ToString());

txtEmail.Text = (string) (rd["Student\_EMAIL"].ToString());

cboOccupation.SelectedText = (string) (rd["Student\_OCCUPATION"].ToString());

cboCourseName.SelectedText = (string) (rd["Student\_CNAME"].ToString());

txtKnowledge.Text = (string) (rd["Student\_KNOWLEDGE"].ToString());

txtAmountPaid.Text = (string) (rd["Student\_TOTALFEES"].ToString());

}

rd.Close();

con.Close();

con.Open();

SqlCommand cmd1 = new SqlCommand("select \* from tbl\_Fee where Fee\_ROLLNO=\'" + str + "\'", con);

SqlDataReader rd1;

rd1 = cmd1.ExecuteReader();

while (rd1.Read())

{

txtReceiptNo.Text = (string) (rd1["Fee\_RECEIPTNO"].ToString());

dtpDate.Text = (string) (rd1["Fee\_Date"].ToString());

txtFee.Text = (string) (rd1["Fee\_BasicFees"].ToString());

txtDiscountPercentage.Text = (string) (rd1["Fee\_DiscountPercentage"].ToString());

txtDicountAmount.Text = (string) (rd1["Fee\_DiscountAmount"].ToString());

txtTotalfee.Text = (string) (rd1["Fee\_TotalFees"].ToString());

txtAmountPaid.Text = (string) (rd1["Fee\_AMOUNT"].ToString());

cboPaymentMode.SelectedText = (string) (rd1["Fee\_ModeOfPayment"].ToString());

txtNameOfBank.Text = (string) (rd1["Fee\_BANKNAME"].ToString());

txtChequeNo.Text = (string) (rd1["Fee\_CHEQUENO"].ToString());

}

rd1.Close();

con.Close();

pnlSearch.Visible = false;

TabControl1.Visible = true;

}

public void btnUpdate\_Click(System.Object sender, System.EventArgs e)

{

con.Open();

string s = "update tbl\_Student set Student\_NAME=\'" + txtStudentName.Text + "\',Student\_FNAME=\'" + txtFatherName.Text + "\',Student\_MNAME=\'" + txtMotherName.Text + "\',Student\_HNAME=\'" + txtHusbandName.Text + "\',Student\_ADDRESS=\'" + txtAddress.Text + "\',Student\_PHONE=\'" + txtPhoneNo.Text + "\',Student\_QUALIFICATION=\'" + cboQualification.Text + "\',Student\_GENDER=\'" + cboGender.Text + "\',Student\_DOB=\'" + dtpDob.Text + "\',Student\_EMAIL=\'" + txtEmail.Text + "\',Student\_OCCUPATION=\'" + cboOccupation.Text + "\',Student\_CNAME=\'" + cboCourseName.Text + "\',Student\_KNOWLEDGE=\'" + txtKnowledge.Text + "\',Student\_TOTALFEES=\'" + txtAmountPaid.Text + "\' where Student\_ROLLNO=\'" + txtRollNo.Text + "\'";

SqlCommand cmd1 = new SqlCommand(s, con);

cmd1.ExecuteNonQuery();

con.Close();

con.Open();

string s1 = "update tbl\_Fee set Fee\_RECEIPTNO=\'" + txtReceiptNo.Text + "\',Fee\_Date=\'" + dtpDate.Value.ToString() + "\',Fee\_BasicFees=\'" + txtFee.Text + "\',Fee\_DiscountPercentage=\'" + txtDiscountPercentage.Text + "\',Fee\_DiscountAmount=\'" + txtDicountAmount.Text + "\',Fee\_TotalFees=\'" + txtTotalfee.Text + "\',Fee\_AMOUNT=\'" + txtAmountPaid.Text + "\',Fee\_ModeOfPayment=\'" + cboPaymentMode.Text + "\',Fee\_BANKNAME=\'" + txtNameOfBank.Text + "\',Fee\_CHEQUENO=\'" + txtChequeNo.Text + "\' where Fee\_ROLLNO=\'" + txtRollNo.Text + "\'";

SqlCommand cmd2 = new SqlCommand(s1, con);

cmd2.ExecuteNonQuery();

MessageBox.Show("Record Updated");

con.Close();

this.Close();

}

public void btnDelete\_Click(System.Object sender, System.EventArgs e)

{

con.Open();

string s = "delete from tbl\_Student where Student\_ROLLNO=\'" + txtRollNo.Text + "\'";

SqlCommand cmd2 = new SqlCommand(s, con);

cmd2.ExecuteNonQuery();

//Else

con.Close();

con.Open();

string s3 = "delete from tbl\_Fee where Fee\_ROLLNO=\'" + txtRollNo.Text + "\'";

SqlCommand cmd3 = new SqlCommand(s3, con);

cmd3.ExecuteNonQuery();

//Else

con.Close();

MessageBox.Show("Record Deleted");

this.Close();

}

public void btnClose\_Click(System.Object sender, System.EventArgs e)

{

pnlSearch.Visible = false;

TabControl1.Visible = true;

}

public void btnExit\_Click(System.Object sender, System.EventArgs e)

{

this.Close();

}

public void cboPaymentMode\_SelectedIndexChanged(System.Object sender, System.EventArgs e)

{

if (cboPaymentMode.SelectedIndex != 0)

{

txtNameOfBank.Enabled = true;

txtChequeNo.Enabled = true;

}

if (cboPaymentMode.SelectedIndex == 0)

{

txtNameOfBank.Enabled = false;

txtChequeNo.Enabled = false;

}

}

public void txtPhoneNo\_KeyPress(System.Object sender, System.Windows.Forms.KeyPressEventArgs e)

{

if ((Microsoft.VisualBasic.Strings.Asc(e.KeyChar) < 48) || (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) > 57))

{

e.Handled = true;

}

if (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) == 8)

{

e.Handled = false;

}

}

public void txtDiscountPercentage\_KeyPress(System.Object sender, System.Windows.Forms.KeyPressEventArgs e)

{

if ((Microsoft.VisualBasic.Strings.Asc(e.KeyChar) < 48) || (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) > 57))

{

e.Handled = true;

}

if (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) == 8)

{

e.Handled = false;

}

}

public void txtTotalfee\_KeyPress(System.Object sender, System.Windows.Forms.KeyPressEventArgs e)

{

if ((Microsoft.VisualBasic.Strings.Asc(e.KeyChar) < 48) || (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) > 57))

{

e.Handled = true;

}

if (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) == 8)

{

e.Handled = false;

}

}

}

}

**Faculty:-**

using System.Diagnostics;

using System;

using System.Xml.Linq;

using System.Windows.Forms;

using System.Collections;

using System.Drawing;

using Microsoft.VisualBasic;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Data.SqlClient;

namespace ComputerManagmentSystem

{

public partial class StaffDetail

{

public StaffDetail()

{

InitializeComponent();

//Added to support default instance behavour in C#

if (defaultInstance == null)

defaultInstance = this;

}

#region Default Instance

private static StaffDetail defaultInstance;

/// <summary>

/// Added by the VB.Net to C# Converter to support default instance behavour in C#

/// </summary>

public static StaffDetail Default

{

get

{

if (defaultInstance == null)

{

defaultInstance = new StaffDetail();

defaultInstance.FormClosed += new FormClosedEventHandler(defaultInstance\_FormClosed);

}

return defaultInstance;

}

}

static void defaultInstance\_FormClosed(object sender, FormClosedEventArgs e)

{

defaultInstance = null;

}

#endregion

SqlConnection con = new SqlConnection("Data Source=.\\sqlexpress;Initial Catalog=D:\\READY PROJECTS\\COMPUTERMANAGMENTSYSTEM\\DB\_CIMS.MDF;Integrated Security=True");

public void loadStaff()

{

con.Open();

SqlCommand cmd = new SqlCommand("select Staff\_NAME from tbl\_Staff", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

cboStaff.Items.Add(rd[0]);

}

con.Close();

}

public void StaffDetail\_Load(System.Object sender, System.EventArgs e)

{

loadStaff();

btnUpdate.Visible = false;

btnDelete.Visible = false;

pnlSearch.Visible = false;

SqlDataAdapter adapter = new SqlDataAdapter("select [ID]=Staff\_No,[Name]=Staff\_NAME,[Qualification]=Staff\_QUALIFICATION,[Contact Number]=Staff\_PHONE from tbl\_Staff", con);

DataSet ds = new DataSet();

adapter.Fill(ds);

dgvSearch.DataSource = ds.Tables[0];

}

public void clear()

{

txtName.Clear();

txtAddress.Clear();

txtEmailId.Clear();

txtExperience.Clear();

txtPhoneNo.Clear();

txtQualification.Clear();

dtpDob.ResetText();

}

public void increment()

{

}

public void btnClear\_Click(System.Object sender, System.EventArgs e)

{

clear();

}

public void btnExit\_Click(System.Object sender, System.EventArgs e)

{

this.Close();

}

public void btnSave\_Click(System.Object sender, System.EventArgs e)

{

if (txtEmailId.Text == "" || txtName.Text == "" || txtAddress.Text == "")

{

Interaction.MsgBox("Details are Incomplete", MsgBoxStyle.Exclamation, null);

}

else

{

con.Open();

string s = "Insert into tbl\_Staff(Staff\_No,Staff\_NAME,Staff\_DOB,Staff\_QUALIFICATION,Staff\_EXPERIENCE,Staff\_EMAILID,Staff\_PHONE,Staff\_ADDRESS) values(\'" + txtStaffNo.Text + "\',\'" + txtName.Text + "\',\'" + dtpDob.Value.ToString() + "\',\'" + txtQualification.Text + "\',\'" + txtExperience.Text + "\',\'" + txtEmailId.Text + "\',\'" + txtPhoneNo.Text + "\',\'" + txtAddress.Text + "\')";

SqlCommand cmd1 = new SqlCommand(s, con);

cmd1.ExecuteNonQuery();

MessageBox.Show("Record Inserted");

clear();

}

}

public void btnSearchUpdate\_Click(System.Object sender, System.EventArgs e)

{

pnlSearch.Visible = true;

btnUpdate.Visible = true;

btnDelete.Visible = true;

btnSave.Visible = false;

btnClear.Visible = false;

pnlStaffDetails.Visible = false;

}

public void dgvSearch\_CellMouseDoubleClick(System.Object sender, System.Windows.Forms.DataGridViewCellMouseEventArgs e)

{

pnlStaffDetails.Visible = true;

con.Open();

string str = dgvSearch.CurrentRow.Cells["ID"].Value.ToString();

SqlCommand cmd = new SqlCommand("select \* from tbl\_Staff where Staff\_No=\'" + str + "\'", con);

SqlDataReader rd;

rd = cmd.ExecuteReader();

while (rd.Read())

{

txtStaffNo.Text = (string) (rd["Staff\_No"].ToString());

txtName.Text = (string) (rd["Staff\_NAME"].ToString());

dtpDob.Text = (string) (rd["Staff\_DOB"].ToString());

txtQualification.Text = (string) (rd["Staff\_QUALIFICATION"].ToString());

txtExperience.Text = (string) (rd["Staff\_EXPERIENCE"].ToString());

txtEmailId.Text = (string) (rd["Staff\_EMAILID"].ToString());

txtPhoneNo.Text = (string) (rd["Staff\_PHONE"].ToString());

txtAddress.Text = (string) (rd["Staff\_ADDRESS"].ToString());

}

rd.Close();

con.Close();

pnlSearch.Visible = false;

pnlStaffDetails.Visible = true;

}

public void cboStaff\_SelectedIndexChanged(System.Object sender, System.EventArgs e)

{

dgvSearch.Columns.Clear();

con.Open();

SqlDataAdapter adapter = new SqlDataAdapter("select [ID]=Staff\_No,[Name]=Staff\_NAME,[Qualification]=Staff\_QUALIFICATION,[Contact Number]=Staff\_PHONE from tbl\_Staff where Staff\_NAME=\'" + cboStaff.Text + "\'", con);

DataSet ds = new DataSet();

adapter.Fill(ds);

dgvSearch.DataSource = ds.Tables[0];

con.Close();

}

public void btnUpdate\_Click(System.Object sender, System.EventArgs e)

{

con.Open();

string s = "update tbl\_Staff set Staff\_NAME=\'" + txtName.Text + "\',Staff\_DOB=\'" + dtpDob.Value.ToString() + "\',Staff\_QUALIFICATION=\'" + txtQualification.Text + "\',Staff\_EXPERIENCE=\'" + txtExperience.Text + "\',Staff\_EMAILID=\'" + txtEmailId.Text + "\',Staff\_PHONE=\'" + txtPhoneNo.Text + "\',Staff\_ADDRESS=\'" + txtAddress.Text + "\' where Staff\_No=\'" + txtStaffNo.Text + "\'";

SqlCommand cmd1 = new SqlCommand(s, con);

cmd1.ExecuteNonQuery();

MessageBox.Show("Record Updated");

con.Close();

this.Close();

}

public void btnDelete\_Click(System.Object sender, System.EventArgs e)

{

con.Open();

string s = "delete from tbl\_Staff where Staff\_No=\'" + txtStaffNo.Text + "\'";

SqlCommand cmd2 = new SqlCommand(s, con);

cmd2.ExecuteNonQuery();

//Else

MessageBox.Show("Record Deleted");

this.Close();

con.Close();

}

public void btnClose\_Click(System.Object sender, System.EventArgs e)

{

pnlSearch.Visible = false;

pnlStaffDetails.Visible = true;

}

public void txtPhoneNo\_KeyPress(System.Object sender, System.Windows.Forms.KeyPressEventArgs e)

{

if ((Microsoft.VisualBasic.Strings.Asc(e.KeyChar) < 48) || (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) > 57))

{

e.Handled = true;

}

if (Microsoft.VisualBasic.Strings.Asc(e.KeyChar) == 8)

{

e.Handled = false;

}

}

}

}

**SYSTEM IMPLEMENTATION**

Specification is also used as a reference point during product implementation.

In fact, the ultimate goal of the implementation is to build a product that need specification.

Thus the implementation use specification during design to make design decision and during the verification activity to check that the implementation complies with specification.

### Implementation Phases:

Once the System is tested satisfactorily with test data, the installation has to be done. This entirely new developed system, will remove the existing manual procedure.

**USER TRAINING:**

* User training is the important factor of system factor installation.
* This will be an ongoing process, which has started during the testing phases.
* The users were involved in the project right from the analysis phase.
* A continuous progress has been show to them from time to time.
* A proper training was given to them before installation of the system.
* For this purpose the test date is entered and various required reports are generated from the system and checked manually.
* After completion of the user training the conversion took place.

**CONVERSION PHASE:**

* Conversion is the process of changing the old system to the new one.
* There are different methods available for conversion.
* In our case even tough there is no existing or automated system, we are using the phase in method.
* The old system is being used along with the new system till the new system can operate without causing major problem.
* This is being done to protect the major loss on the business.

**POST IMPLEMENTATION PHASE:**

* After the successful installation of newly developed system, the important step in the system development life cycle is post installation review.
* The success of the system will be judged by the review.
* The outputs are the real objectives of the system and unless they are correct, the installation is not correct.
* The frequency of the output varies and thus each of them must be tested at the particular frequency.
* Therefore post installation review may continue for 2 to 4 months.
* We have decided to keep track of the system for the next 4 months, which will help us to gather the information for the maintenance of the system.
* In post implementation review we will come to know the various problem face by the user in handling the new system.
* This information will be very useful to make necessary changes in the new system.

**Maintenance**

**Security:**

* **Tours and Travels** makes use of security in the form of User ID’s and Passwords.
* In order to use this system, the administrator has to enter his password. The password entered by the administrator is compared with the one in the database and if it matches then the application precedes it to open.
* At no point in time the administrator can recover his password if he forgets it. However it is recommended that the administrator change his /her password in the database.
* For security purposes it is recommended that the administrator change his password every 2 weeks.
* Besides this on the backend system makes use of Microsoft Access, which is very secure DBMS. At all point in the DBMS checks the username and password before making a connection.’ If the username and passwords match then the connection is allowed else it is not allowed.

**SYSTEM REQUIREMENTS**

**Software Requirement:**

* Operating System : Microsoft Windows XP
* Front End: : Visual studio 2008
* Back End: : MS SQL server 2005

**Hardware Requirement:**

Minimum Requirements:

* P3.
* 450 MHz.
* 10 GB Hard Disk.
* 128 Mb Ram.

Recommended:

* P4.
* 2.4 GHz.
* 20 GB Hard Disk.
* 256 Mb Ram.
* Operating System (OS) – Windows XP.

**CONCLUSION**

The demand of Visual Studio for application is increasing day by day in Software industry, due to high expectations of client companies. Hence an attempt of automating an office application had added to out learning experience.

It has also helped in adopting an analytical approach to solving and made us realize that system development is a step by step process, thereby appreciating the role of SDLC model in organizing the complex process of system development into manageable chunks. Indeed it was a great learning experience.

**Future Enhancement**

This project can be further enhanced to provide greater flexibility and performance with certain modification whenever necessary

**BIBLIOGRAPHY**

**References:-**

* Murach - Beginning ASP .NET
* Murach - Visual Basic 2008
* System analysis & design in a changing world Software Engineering by Roger.S.Pressman
* [www.w3schools.com](http://www.w3schools.com)
* [www.google.com](http://www.google.com)