

Electronic City Phase 2, Opp.HCL, Bangalore-100

A Project Report On

Virtual Campus

College Management System

Submitted in Partial Fulfilment of the Requirement Of the 6th Semester of “Bachelor of Computer Applications” **(BCA)** By Bangalore University, During the Academic year 2015-2016.

Under The Guidance of

**Prof**. **Balakrishna**

**(HOD OF BCA)**

Submitted By:

**MANJUNATH M (124KSB6010)**



IZEE

COLLEGE OF MANAGEMENT

AND INFORMATION SCIENCE

Electronic City Phase 2, Bangalore-100

**A PROJECT REPORT ON**

Virtual Campus

COLLEGE MANAGEMENT SYSTEM

Submitted In partial Fulfilment of the Requirement For the 6th Semester of “Bachelor of Computer Applications” (BCA) By Bangalore University, During the Academic Year 2014-2015.

Under The Guidance Of

**Prof. BALAKRISHNA**

(HOD of BCA)

Submitted by:

**MANJUNATH M (124KSB6010)**



CERTFICATE

This Is To certify, **MANJUNATH M (124KSB6010) of The 6th Semester BCA**

**(BACHELOR OF COMPUTER APPLICATIONS),** has satisfactorily completed

The Software Engineering Lab Project entitled **“VIRTAUAL CAMPUS”** **The**

**College Management System** as Prescribed by the Bangalore University

From Our Institution During The year 2015-2016.

**Lecture in Charge Head Of The Department**

**External Examiners**

**PLACE:**

**DATE:**

DECLARATION

Me the Student Of 6th semester BCA**, IZEE college of management and**

**Information science,** Bangalore, Hereby declare that dissertation entitled

**“VIRTUAL CAMPUS-The College management System** “Has been developed

By the Individual Member **MANJUNATH M** for the partial fulfilment of the requirements for the 6th Semester of **BACHELOR OF COMPUTER APPLICATIONS**

By the BangaloreUniversity during the Academic Year 2015-2016.

Further, The Matter for Embodied in The Dissertation has not been submitted

Previously by anybody for the award of any degree or diploma to any other

University.

DATE:

PLACE: BANGALORE MANJUNATH M

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We Extend Our Gratitude towards our Head of the BCA Department And

Our Project Guide Prof. MALLINATH for his valuable guidance throughout

The Course of This Project.

I Am Thankful to each and every person who has helped in the Successful

Completion of This Project.

Thankfully,

**MANJUNATH M**

CONTENTS

CONTENTS

1. ABSTRACT
2. INTRODUCTION
3. OBJECTIVES
4. TOOLS/ENVIRONMENT USED
5. ANALYSIS DOCUMENT
6. DESIGN DOCUMENT
7. PROGRAM CODE
8. TESTING
9. INPUT/OUTPUT SCREENS
10. CONCLUSION/FUTURE APPLICATION/ENHANCEMENT OF THE PROJECT
11. BIBLIOGRAPGY

ABSTRACT

ABSTRACT

This Project is aimed to help the college institutions to overcome the problems they

Face in day to day life including the manually operational overhead work due to which

The documents of the institutions cannot be processed for approvals and the problems

Of managing student database, employees database, student scholarship database,

Accounts Management, Payroll management, Student management etc…

This Web Application Software Is Fully Integrated With **SRM** (**Student Relationship Management)** as well as **CMS (Content Management System)** Solution.

This Software Is developed in Such a Manner that is easily manageable, Time saving

And Mainly Relieving One from Manual Works.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of **MS-SQLServer** and all the user interfaces have been designed using the ASP.Net technologies. The database connectivity is planned using the “SQL Connection” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specifications have been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are

Executed by the general users and the organizational administration. The user interfaces

Are browser specific to give distributed accessibility for the overall system, the internal Database has been selected as MS-SQL server. The basic constructs of table spaces,

Clusters and indexes have been exploited to provide higher consistency and reliability for

The data storage. The MS-SQL server was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the ASP.Net technologies. At all proper levels high care was taken to check that the system manages

The data consistency with proper business rules or validations. The database connectivity was planned using the latest ”SQL Connection” technology provided by **©Microsoft Corporation**. The authentication and authorization was crosschecked at all the relevant stages.

INTRODUCTION

INTRODUCTION

The Implementation of computers in each and every field of life made the

Life much easier. The work became more efficient, accurate & less complex.

The project has been planned to be having the view of distributed architecture,

With centralized storage of the database. The application for the storage of the

Data has been planned. Using the constructs of MS-SQLServer and all the user interfaces have been designed using the ASP.Net technologies. The database connectivity is planned using the “SQL Connection” methodology. The standards

Of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

Main Idea of my Project Is To Connect Student & College Anytime, Anywhere

All 24/7 and 365 Days A Year, This Is the Network of Connections Only

Between The Administrator, Accounts Department, Students of the Institution.

My Project Comes Under The category Of Web Application Programs in Which I

Tries to reduce the human efforts in “**COLLEGE CAMPUS**” and creating

Smooth flow of work Anytime Anywhere Using Proper Authentication.

It Would Be Really My Pleasure to Introduce a New Feature Where I tried my

Hardest To Make the Communication between the College & the Students

Anywhere Anytime On the Go.

This Project Is Meant For Creating a Web Application Software to Do the Various

College Management Tasks. It Covers Areas like Registering Student, Storing

The Student Information, Updating Student Details, Keeping Track of Student

Details, Collecting Fee and Keeping Track Of Fee Collected in Order to Make

Audits Easy, Managing Student Scholarships, Using Student Login Student Can

Get Access to the Features Provided by the Software and That Access Provides

Safest, Secure Access to His Own Database without Conflicting the Other.

Enhanced Administrative Features and Enhanced Security for a Software And

Database, Facility for the One Click WWW Connect & Made Scholarship Registration

Easy & Desirably Management Of Scholarship Is Very Easy.

OBJECTIVES

OBJECTIVES

* To create Virtual Presence of College Anytime Anywhere.
* The Aim of This Project Is to Create Separate Network And Environment

For The Particular Institutions to Build the Connection between the Student

And The College by Making Separate User Authentications to Access The

Network and to Enjoy the Features Given To the Student Anytime Anywhere

24/7 Online.

* To Computerize the recording, Processing and Documentation of student data in college.
* To Computerize The Payroll Management And Accounts department For The

Reliable and fast, Secure Medium of Transactions.

* To Computerize The Scholarship Management System.
* To Computerize The Notifications Of The Colleges.
* To Computerize Document Collections Of Students By Admin.
* To Computerize Manual Leave Letters.
* To Computerize Notifications By Colleges.
* To Computerize Holiday Settings.
* To Computerize Administrative Tasks.
* To Computerize Accounts Department Tasks.
* To Computerize Student Tasks.

TOOLS/ENVIRONMENT USED

TOOLS/ENVIRONMENT USED

* Software Tools Used :
* **Operating System :**
* Windows 8.1
* **Front End Tool :**
* Microsoft Visual Studio 2010(Service Pack 1)
* Microsoft Visual Web Developer 2010 Express
* HTML 5
* CSS 3.0
* **Back End Tool :**
* Microsoft SQL Server 2008 R2
* Hardware Tools Used :
* CPU : Dual Core
* Memory : 2GB
* Hard Disk : 320 GB

ANALYSIS DOCUMENT

SRS DOCUMENT

1. **Specific Requirements**

* This project is aimed to developing an Virtual college management system
* The entire project has been developed keeping in the view of the distributed client server computing technology in mind.
* Virtual Campus- College Management System is to create an e-Campus virtual Experience between College and the Students and related to Computerized Data Processing.
* Through This Application Administrator Can Do Various Administrative Tasks Built Into This Web Application And Accounts Department Admin Can Do Various Accounts Related Tasks On The Go, Student Can login To This Web Application To Meet His Requirements Which Belongs To That Student & College.
* Moreover If Anyone Parents or General Public Want To Access the Site They Can See The information On the Homepage Site.
* Admin Is the Main Authority to Grant Access to The Services Provide by This Web Application to the Various Groups like Accounts department, Students. And Also Admin Is the One Who Add, Delete, Modify, Track Logins, Upload Notifications, Receive & Reply to Complaints etc…..
* New features To Students Includes Posting New Complaints, Requesting e-Holiday, Downloading Notifications & Uploading Documents required by College etc…

**Existing System**

The Current System works through A Single System, Where A Client makes Specific

Requests.

**Limitations in Existing Systems:-**

* It is limited to a single system.
* It is less User-friendly.
* It is having lots of manual work (Manual system does not mean that you are working with pen and paper, it also include working on spread sheets and other simple software's).
* The present system is very less secure.
* It is unable to generate different kinds of report.
* It doesn’t have the mail and file upload feature.
* Sometimes Calculations May Wrong While Doing Manually.

**Solutions to These Problems:-**

* The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.
* User friendliness is provided in the application with various controls.
* The system makes the overall project management much easier and flexible.
* It can be accessed over the Internet.
* Various classes have been used to provide file upload and mail features.
* Various classes have been used to provide file upload and mail features
* There is no risk of data mismanagement at any level while the project development is under process.
* Report generation feature is provided using Crystal Reports to generate different kinds of reports like bar graphs, pie charts and table type charts etc.
* It provides high level of security using different protocols like https etc.

1. **Feasibility Study**

**Economic feasibility:-**

* Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.
* A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**Operational feasibility:-**

* Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:
* Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
* Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems

**Operational Feasibility:-**

* Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, .at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.
* Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?

|  |
| --- |
| **Software Requirements** |
| Operating System : Windows XP/2003/7/8/8.1 |
| User Interface : HTML 5, CSS 3.0 |
| Client-side Scripting : JavaScript/VBScript |
| Programming Language : Visual C#.net |
| Web Applications : ASP.net |
| Database : SQL Server 2008 R2 |
| **Hardware Requirements** |
| Processor : Pentium IV |
| Hard Disk : 40GB or more |
| RAM : 256MB or more |
|  |

DESIGN METHODOLOGY

**Design Methodologies & Principles**

To produce the design for the large module can be extremely complex task. The design principles are used to provide effective handling the complexity of the design process. It will not reduce to the effort needed for design but can also reduce the scope of introducing errors during design.

For solving the large problems, the problem is divided into smaller pieces, using the time-tested principles of “Divide and Conquer”. This system problem divides into smaller pieces, so that each piece can be conquered separately. For software design, the problem is to divide into manageable small pieces that can be solved separately. This divide principle is used to reduce the cost of the entire problem that means the cost of solving the entire problem is more than the sum of the cost of solving all the pieces.

When partitioning is high, then also arises a problem due to the cost of partitioning. In this situation to know the judgment about when to stop partitioning.

In design the most important quality criteria are simplicity and understandability. In this the part is easily related to the application and that each piece can be modified separately. Proper partitioning will make the system to maintain by making the designer to understand problem partitioning also aids design verification.

Abstraction is an essential for the problem partitioning and is used for adjusting components as well as components that are being designed; abstraction of existing component plays an important role in the maintenance phase during design process of the system.

In the functional abstraction, the main four modules to taking the details and computing for further actions. In data abstraction it provides some services.

The system is a collection of modules means components. The highest- level component corresponds to the total system. For design the system, first following the top down approach to divide the problem in modules. In top down design methods often results in some form of stepwise refinement after divide the main module, the bottom up approach is allowed to designing the most basic or primitive components to higher- level components. The bottom-up method operations starting from very bottom.

DATA DICTIONARY

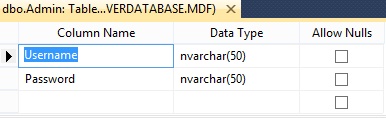
DATA DICTIONARY

After carefully understanding the requirements of the client the entire data storage requirements

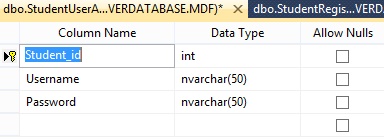
Are divided into tables. The below tables are normalized to avoid any anomalies during the course

Of data entry.

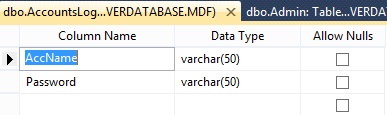
dbo.AdminLogin

****

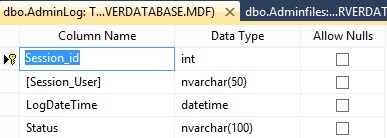
dbo.StudentUserAccounts

****

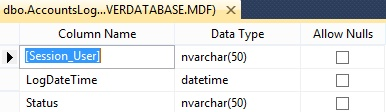
dbo.AccountsLogin

****

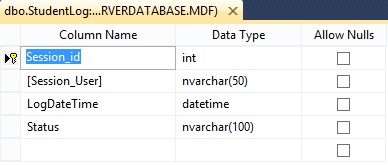
dbo.AdminLog

****

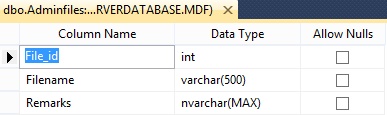
dbo.AccountsLog



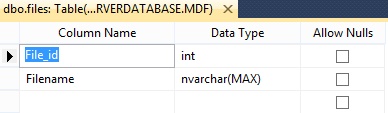
dbo.StudentLog



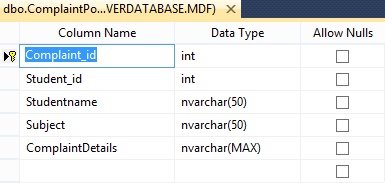
dbo.Adminfiles



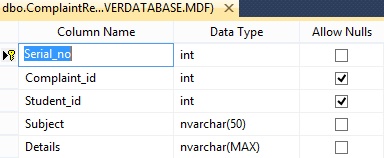
dbo.files



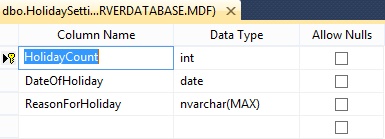
dbo.ComplaintPosting



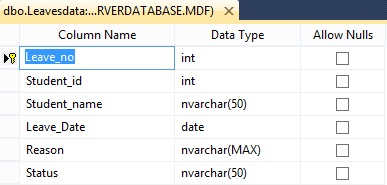
dbo.ComplaintReply



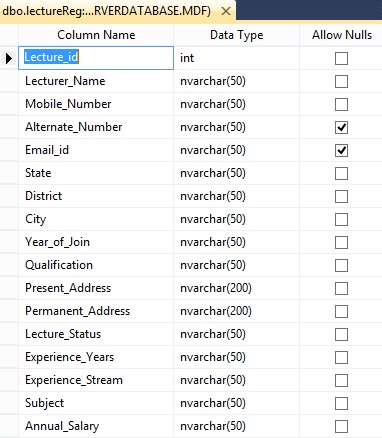
dbo.HolidaySettings



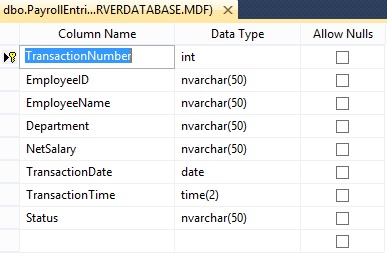
dbo.Leavesdata



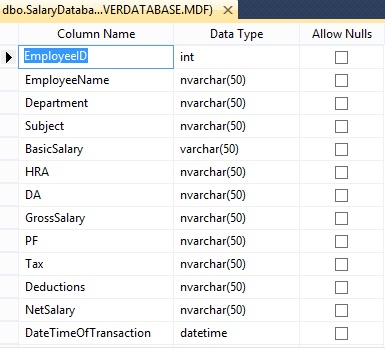
dbo.LectureReg



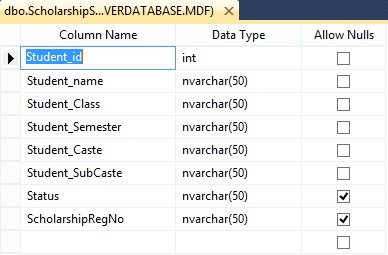
dbo.PayrollEntries



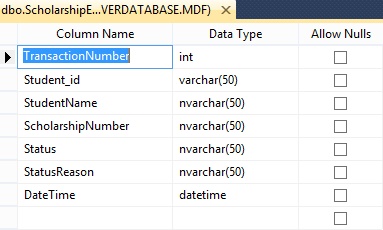
dbo.SalaryDatabase



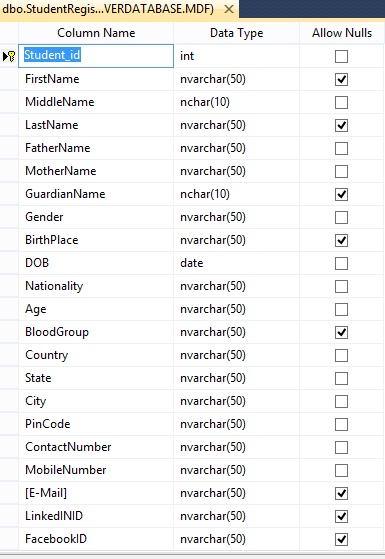
dbo.ScholarshipStat

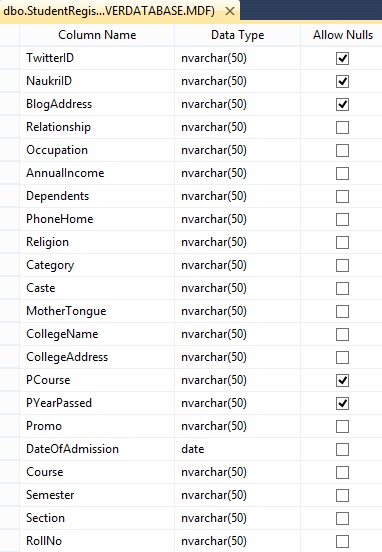


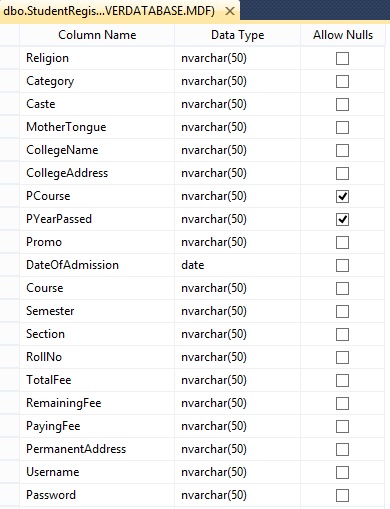
dbo.ScholarshipEntries



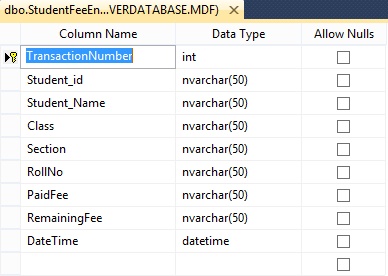
dbo.StudentRegistrations



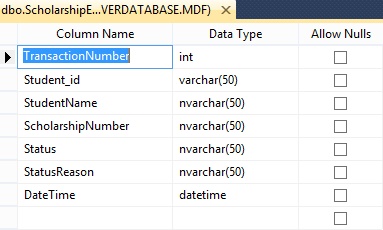




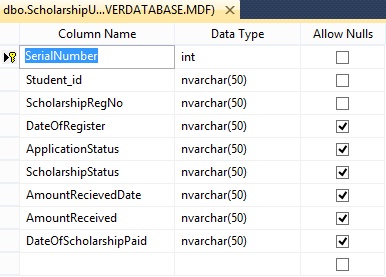
dbo.StudentFeeEntries



dbo.ScholarshipEntries



dbo.ScholarshipUpdate



DESIGN DOCUMENT

DESIGN DOCUMENT

**Study of the System**

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI’S at the top level have been categorized

As:

**1. Administrative user interface**

**2. The operational or generic user interface**

**1. Administrative user interface:**

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities**.**

**2. The operational or generic user interface:**

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

**Component Details**

**Number of Modules:-**

The system after careful analysis has been identified to be presented with the following modules:

1. College Administration.
2. Student Administration.
3. Accounts Administration.
4. Scholarship Management.
5. Payroll Management.
6. Login Trackers.
7. Reports.
8. Authentication.

DATABASE DESIGN

DATABASE DESIGN

**Introduction:**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**NORMALIZATION**

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

**Insertion anomaly**: Inability to add data to the database due to absence of other data.

**Deletion anomaly**: Unintended loss of data due to deletion of other data.

**Update anomaly**: Data inconsistency resulting from data redundancy and partial update

**Normal Forms**: These are the rules for structuring relations that eliminate anomalies.

**FIRST NORMAL FORM:**

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**SECOND NORMAL FORM:**

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

* Primary key is a not a composite primary key
* No non key attributes are present
* Every non key attribute is fully functionally dependent on full set of primary key.

**THIRD NORMAL FORM:**

A relation is said to be in third normal form if their exits no transitive dependencies.

**Transitive Dependency**: If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

PROGRAM CODE

PROGRAM CODES

1. User-Defined Stored Procedure

|  |
| --- |
| ALTER Procedure FindString(@MyString nvarchar(50)) |
| As |
| Begin |
| Select \* From Salarydatabase Where EmployeeName=@Mystring |
| End |
|  |

namespace VirtualCampus

{

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

{

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

{

}

public class ReadData

{

public bool FindString(string myStrings)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "FindString";

cmd.Parameters.AddWithValue("@MyString", myStrings);

try

{

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

return true;

}

}

catch (Exception ex)

{

}

finally

{

if (con.State == ConnectionState.Open)

con.Close();

}

return false;

}

}

1. **Admin login C# Code**

namespace VirtualCampus

{

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

{

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

{

txtUsername.Focus();

}

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

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

string status;

status = "User Logged In Successfully";

string sql, sql2;

sql = "select \*from Admin where Username ='" + txtUsername.Text + "' and Password = '" + txtPassword.Text + "'";

sql2 = "insert into AdminLog([Session\_User],[LogDateTime],[Status]) values('" + txtUsername.Text + "','" + DateTime.Now.ToString() + "','" + status + "')";

cmd.CommandText = sql + sql2;

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

Session["NEW"] = txtUsername.Text;

Response.Redirect("Interface.aspx");

}

else

literromsg.Text = "Can't Login Invalid User!";

}

}

}

1. **Accounts Login C# Code**

namespace VirtualCampus

{

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

{

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

{

}

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

{

txtUsername.Text = "";

txtPassword.Text = "";

}

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

{

Response.Redirect("Default.aspx");

}

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

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

string status;

status = "User Logged In Successfully";

string sql, sql2;

sql = "select \*from AccountsLogin where AccName ='" + txtUsername.Text + "' and Password = '" + txtPassword.Text + "'";

sql2 = "insert into AccountsLog([Session\_User],[LogDateTime],[Status]) values('" + txtUsername.Text + "','" + DateTime.Now.ToString() + "','" + status + "')";

cmd.CommandText = sql + sql2;

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

Session["NEW"] = txtUsername.Text;

Response.Redirect("AccountsHome.aspx");

}

else

lblWhatHappened.Text = "Can't Login...Invalid User!";

}

}

}

**3.Student Login C# Code**

namespace VirtualCampus

{

public partial class WebForm1 : System.Web.UI.Page

{

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

{

}

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

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

string status;

status = "Student Logged In Successfully";

string sql;

sql = "select \*from StudentRegistrations where Student\_id='"+txtStudent\_id.Text+"'";

cmd.CommandText = sql;

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

string User=txtStuUsername.Text.ToString();

string pass = txtStuPassword.Text.ToString();

if (dr.Read())

{

if ((User == dr["Username"].ToString()) && (pass== dr["Password"].ToString()))

{

con.Close();

string sql1;

status = "Student Logged In Successfully";

sql1 = "insert into StudentLog([Session\_User],[LogDateTime],[Status]) values('" + txtStuUsername.Text + "','" + DateTime.Now.ToString() + "','" + status + "')";

cmd.CommandText = sql1;

con.Open();

SqlDataReader dr1 = cmd.ExecuteReader();

Session["NEW"] = txtStudent\_id.Text;

Response.Redirect("StudentInterface.aspx");

con.Close();

}

}

else

lblLoginFailured.ForeColor = System.Drawing.Color.IndianRed;

lblLoginFailured.Text = "This Login Is Incorrect According to Automated Security System! Please Retry...";

lblLoginFailured1.ForeColor = System.Drawing.Color.ForestGreen;

lblLoginFailured1.Text = "Please Register With Admin Before Logging In!";

}

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

{

txtStudent\_id.Text = "";

txtStuUsername.Text = "";

txtStuPassword.Text = "";

}

}

}

4.Change Password C# Code

namespace VirtualCampus

{

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

{

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

{

txtSearch.Focus();

}

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

{

lblPassChanged.Text = "";

txtNewPass.Text = "";

txtRePass.Text = "";

lblCurrentUser.Text = "";

try

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string sql;

sql = "select \* from StudentRegistrations where Student\_id ='" + txtSearch.Text + "'";

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblinfo.ForeColor = System.Drawing.Color.Green;

lblinfo.Text = "Details Found! Below Are The Details";

txtNewPass.Enabled = true;

txtRePass.Enabled = true;

txtStuID.Text = dr["Student\_id"].ToString();

txtStuName.Text = dr["FirstName"].ToString();

txtStuClass.Text = dr["Course"].ToString();

txtYear.Text = dr["Semester"].ToString();

txtSection.Text = dr["Section"].ToString();

txtOldPwd.Text = dr["Password"].ToString();

lblCurrentUser.Text = dr["Username"].ToString();

}

else

{

txtNewPass.Enabled = false;

txtRePass.Enabled = false;

lblinfo.ForeColor = System.Drawing.Color.DarkRed;

lblinfo.Text = "Student Details Doesn't Exists! Please Review And Try Again";

txtStuID.Text = "";

txtStuName.Text = "";

txtStuClass.Text = "";

txtYear.Text = "";

txtSection.Text = "";

txtOldPwd.Text = "";

txtSearch.Text = "";

}

}

catch (SqlException e1)

{

string display = ("Message From Manjunath:" + e1);

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

}

}

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

{

string newPassword = txtNewPass.Text.Replace("'", "''");

string Repass = txtRePass.Text.Replace("'", "''");

if (txtNewPass.Text != txtRePass.Text)

{

lblPassChanged.Text = "Passwords Do Not Match!";

txtNewPass.Text = "";

txtRePass.Text = "";

}

else

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string sql = "select \* from StudentRegistrations where Student\_id='" + txtStuID.Text + "' and FirstName='" + txtStuName.Text + "' and Username='" + lblCurrentUser.Text + "' and Password ='" + txtOldPwd.Text + "'";

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

dr.Close();

string sqll = "update StudentRegistrations set Password='" + Repass + "' where Student\_id='" + txtStuID.Text + "' and FirstName='" + txtStuName.Text + "' and Username='" + lblCurrentUser.Text + "' and Password ='" + txtOldPwd.Text + "'";

SqlCommand cmd1 = new SqlCommand(sqll, con);

int rows = cmd1.ExecuteNonQuery();

lblPassChanged.Text = " Password Changed Successfully";

con.Close();

}

}

}

}

}

5. Admin Password Change C# Code

namespace VirtualCampus

{

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

{

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

{

if (Session["NEW"] != null)

{

lblCurrentAdmin.Text = Session["NEW"].ToString();

}

}

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

{

txtoldPwd.Text = "";

txtNewPwd.Text = "";

txtRnPwd.Text = "";

}

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

{

string newPassword = txtNewPwd.Text.Replace("'", "''");

string Repass = txtRnPwd.Text.Replace("'", "''");

if (txtNewPwd.Text != txtRnPwd.Text)

{

lblPassChanged.Text = "Passwords Do Not Match! Please Verify And try Again";

txtNewPwd.Text = "";

txtRnPwd.Text = "";

}

else

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string sql = "select \* from Admin where Username='" + lblCurrentAdmin.Text + "' and Password='" + txtoldPwd.Text + "'";

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

dr.Close();

string sqll = "update Admin set Password='" + txtRnPwd.Text + "' where Username='" + lblCurrentAdmin.Text + "'";

SqlCommand cmd1 = new SqlCommand(sqll, con);

int rows = cmd1.ExecuteNonQuery();

lblPassChanged.Text = "Password Changed Successfully";

txtoldPwd.Text = "";

txtNewPwd.Text = "";

txtRnPwd.Text = "";

con.Close();

}

else

{

lblPassChanged.Text = "Old Password Entered Was Wrong";

}

}

}

}

}

6. Holiday Settings C# Code

namespace VirtualCampus

{

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

{

private DataView eventData;

private DataView EventData

{

get

{

if (eventData == null)

{

eventData =

SqlDataSource1.Select(new DataSourceSelectArguments()) as DataView;

}

return eventData;

}

set

{

eventData = value;

}

}

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

{

}

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

{

try

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "insert into HolidaySetting([DateOfHoliday],[ReasonForHoliday]) values('" + Calendar1.SelectedDate + "','" + txtReason.Text + "')";

cmd.ExecuteNonQuery();

lblUpdateStatus.ForeColor = System.Drawing.Color.DarkGreen;

lblUpdateStatus.Text = "Holiday Successfully Hosted!";

GridView1.DataBind();

con.Close();

}

catch(SqlException err)

{

lblUpdateStatus.Text = "Server Error! Holiday Cannot Be Hosted :( Please Try After Sometimes";

}

}

protected void Calendar1\_DayRender(object sender, DayRenderEventArgs e)

{

if (EventData.Count > 0)

{

System.DateTime testDate;

foreach (DataRowView testRow in EventData)

{

testDate = (System.DateTime)testRow["DateOfHoliday"];

if (testDate.Date == e.Day.Date)

{

e.Cell.BackColor = System.Drawing.Color.Red;

}

}

}

}

}

}

7.Leave Reply C# Code

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

string acc = "accepted";

string rej = "rejected";

string str = "Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

SqlConnection con = new SqlConnection(str);

if (e.CommandName == "Accept")

{

con.Open();

string q = "update Leavesdata set Status=@Status where Leave\_no=@Leave\_no and Student\_id=@Student\_id and Student\_name=@Student\_name";

int index = Convert.ToInt32(e.CommandArgument);

GridViewRow row = GridView1.Rows[index];

string Leave\_no = row.Cells[1].Text.ToString();

string Student\_id = row.Cells[2].Text.ToString();

string Student\_name = row.Cells[3].Text.ToString();

string Status = row.Cells[4].Text.ToString();

SqlCommand cmd = new SqlCommand(q, con);

cmd.Parameters.AddWithValue("@Leave\_no", Leave\_no);

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

cmd.Parameters.AddWithValue("@Student\_name", Student\_name);

cmd.Parameters.AddWithValue("@Status", acc);

cmd.ExecuteNonQuery();

lblWhatHappened.ForeColor = System.Drawing.Color.DarkGreen;

lblWhatHappened.Text = "Accepted Successfully By Admin And Informed To Student..Thank You For Using This Service";

GridView1.DataBind();

}

if (e.CommandName == "Reject")

{

con.Open();

string q = "update Leavesdata set Status=@Status where Leave\_no=@Leave\_no";

int index = Convert.ToInt32(e.CommandArgument);

GridViewRow row = GridView1.Rows[index];

string Leave\_no = row.Cells[1].Text.ToString();

string Student\_id = row.Cells[2].Text.ToString();

string Student\_name = row.Cells[3].Text.ToString();

string Status = row.Cells[4].Text.ToString();

SqlCommand cmd = new SqlCommand(q, con);

cmd.Parameters.AddWithValue("@Leave\_no", Leave\_no);

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

cmd.Parameters.AddWithValue("@Student\_name", Student\_name);

cmd.Parameters.AddWithValue("@Status", rej);

cmd.ExecuteNonQuery();

lblWhatHappened.ForeColor = System.Drawing.Color.IndianRed;

lblWhatHappened.Text = "Rejected Successfully By Admin And Informed To Student..Thank You For Using This Service";

GridView1.DataBind();

}

}

}

}

1. Lecture payroll Manager C# Code

namespace VirtualCampus

{

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

{

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

{

}

public class ReadData

{

public bool FindString(string myStrings)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "FindString";

cmd.Parameters.AddWithValue("@MyString", myStrings);

try

{

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

return true;

}

}

catch (Exception ex)

{

}

finally

{

if (con.State == ConnectionState.Open)

con.Close();

}

return false;

}

}

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

{

try

{

if (lblEmpID.Text == "" && lblEmpName.Text == "")

{

string display = "Please Select Emplyee From The List to Calculate Salary Breakups!";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

Button2.Enabled = false;

}

string myString = lblName.Text;

ReadData r = new ReadData();

if (r.FindString(myString) == true)

{

lblWhatHappened.ForeColor = System.Drawing.Color.Red;

lblWhatHappened.Text = "Record Already Exists";

Button2.Enabled = false;

Button3.Enabled = false;

}

else

{

lblWhatHappened.ForeColor = System.Drawing.Color.DarkGreen;

lblWhatHappened.Text = "Available To Record!";

Button2.Enabled = true;

Button3.Enabled = true;

Button1.Enabled = false;

Button2.Enabled = true;

string empname = lblName.Text;

string empno = lblID.Text;

float basic = Convert.ToInt32(txtCalculateSalary.Text);

float hra = Convert.ToInt32(basic \* 0.4);

float da = Convert.ToInt32(basic \* 0.6);

float gross = Convert.ToInt32(basic + hra + da);

float pf = Convert.ToInt32(gross \* 0.13);

float tax = Convert.ToInt32(gross \* 0.2);

float deductions = Convert.ToInt32(pf + tax);

float netsalary = Convert.ToInt32(gross - deductions);

lblEmpBasic.Text = basic.ToString(); ;

lblHRA.Text = hra.ToString();

lblDA.Text = da.ToString();

lblGross.Text = gross.ToString();

lblPF.Text = pf.ToString();

lblTax.Text = tax.ToString();

lblDeductions.Text = deductions.ToString();

lblTotal.Text = netsalary.ToString();

}

}

catch (SqlException myex)

{

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myex + "');", true);

Button2.Enabled = false;

}

}

protected void Button3\_Click(object sender, EventArgs e)

{

if (Button1.Enabled == true)

{

Button1.Enabled = false;

}

else if(Button1.Enabled==false)

{

Button1.Enabled = true;

}

}

protected void ListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

Button1.Enabled = true;

Button2.Enabled = false;

string jesus = ListBox1.SelectedItem.Value;

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

string sql;

sql = "select \* from lectureReg where Lecture\_id ='" + jesus + "'";

cmd.CommandText = sql;

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblID.Text = dr["Lecture\_id"].ToString();

lblName.Text = dr["Lecturer\_Name"].ToString();

lblDept.Text = dr["Experience\_Stream"].ToString();

lblSubject.Text = dr["Subject"].ToString();

lblEmpID.Text = lblID.Text;

lblEmpName.Text = lblName.Text;

}

else

{

dr.Close();

Button1.Enabled = false;

Button2.Enabled = false;

lblID.Text = "";

lblName.Text = "";

lblDept.Text = "";

lblSubject.Text = "";

}

}

catch (IndexOutOfRangeException ex)

{

ListBox1.ClearSelection();

Response.Write("Message From Manjunath:" + ex);

}

catch (SqlException sql)

{

Response.Write("Message From Manjunath:"+sql);

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

if (lblID.Text == "" && lblName.Text == "")

{

string display = "Please Select Emplyee From The List to Assign Salary";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

Button2.Enabled = false;

}

else if (lblEmpName.Text == "" && lblEmpName.Text == "")

{

string display = "Please Select Emplyee From The List to Assign Salary";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

Button2.Enabled = false;

}

else

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "insert into Salarydatabase([EmployeeID],[EmployeeName],[Department],[Subject],[BasicSalary],[HRA],[DA],[GrossSalary],[PF],[Tax],[Deductions],[NetSalary],[DateTimeOfTransaction]) values('" + lblEmpID.Text + "','" + lblEmpName.Text + "','" + lblDept.Text + "','" + lblSubject.Text + "','" + lblEmpBasic.Text + "','" + lblHRA.Text + "','" + lblDA.Text + "','" + lblGross.Text + "','" + lblPF.Text + "','" + lblTax.Text + "','" + lblDeductions.Text + "','" + lblTotal.Text + "','" + DateTime.Now.ToString() + "')";

cmd.ExecuteNonQuery();

lblWhatHappened.Text = "Transaction Success";

cmd.CommandText = "insert into PayrollEntries([EmployeeID],[EmployeeName],[Department],[NetSalary],[TransactionDate],[TransactionTime],[Status]) values('" + lblEmpID.Text + "','" + lblEmpName.Text + "','" + lblDept.Text + "','" + lblTotal.Text + "','" + DateTime.Today.ToShortDateString() + "','" + DateTime.Now.ToLocalTime() +"','" + lblWhatHappened.Text + "')";

cmd.ExecuteNonQuery();

SqlDataReader dr = cmd.ExecuteReader();

lblWhatHappened.Text = "Transaction Success";

dr.Close();

con.Close();

}

}

catch (SqlException exe)

{

lblWhatHappened.ForeColor = System.Drawing.Color.Red;

lblWhatHappened.Text = "Transaction Failed!";

Response.Write("Message From Manjunath:" + exe);

}

}

protected void Button4\_Click(object sender, EventArgs e)

{

ListBox1.ClearSelection();

Response.Redirect("AccountsHome.aspx");

}

}

10.Request e-Holiday C# Code

namespace VirtualCampus

{

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

{

private DataView eventData;

private DataView EventData

{

get

{

if (eventData == null)

{

eventData =

SqlDataSource1.Select(new DataSourceSelectArguments()) as DataView;

}

return eventData;

}

set

{

eventData = value;

}

}

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

{

if (Session["NEW"] != null)

{

lblUser.Text = Session["NEW"].ToString();

}

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string sql;

sql = "select \* from StudentRegistrations where Student\_id ='" + lblUser.Text + "'";

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblWhatHappened.Text = "$Connected To $Server";

lblStuName.Text = dr["FirstName"].ToString();

}

else

{

con.Close();

dr.Close();

lblWhatHappened.ForeColor = System.Drawing.Color.IndianRed;

lblWhatHappened.Text = "Can't Start Complaint Post Service! Please Try Later";

Button1.Enabled = false;

}

}

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

{

try

{

string status = "Pending";

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "insert into Leavesdata([Student\_id],[Student\_name],[Leave\_Date],[Reason],[Status]) values('" + lblUser.Text + "','" + lblStuName.Text + "','" + Calendar1.SelectedDate + "','" + TextBox1.Text + "','" + status + "')";

cmd.ExecuteNonQuery();

lblWhatHappened.ForeColor = System.Drawing.Color.DarkGreen;

lblWhatHappened.Text = "Leave Request Submission Successful";

GridView1.DataBind();

con.Close();

}

catch (SqlException err)

{

Response.Write("Message :" + err);

lblWhatHappened.Text = "Server Error!Please Try After Sometimes";

}

}

protected void Calendar1\_DayRender(object sender, DayRenderEventArgs e)

{

try

{

if (EventData.Count > 0)

{

System.DateTime testDate;

foreach (DataRowView testRow in EventData)

{

testDate = (System.DateTime)testRow["Leave\_Date"];

if (testDate.Date == e.Day.Date)

{

e.Cell.BackColor = System.Drawing.Color.Red;

}

}

}

}

catch (NullReferenceException exe)

{

Response.Write("Message from manjunath :" + exe);

}

}

}

}

1. Scholarship Transaction C# Code

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

{

try

{

if (lblid.Text == "" && lblname.Text == "")

{

string display = "Please Select Student From The List!";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

cmdSave.Enabled = false;

}

else if (lblclass.Text=="" && lblsemester.Text =="")

{

string display = "Select Student First!";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

cmdSave.Enabled = false;

}

else

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update ScholarshipStat SET Status='" + ddlDispenseStatus.Text + "', ScholarshipRegNo='" + txtregno.Text + "' where Student\_id='" + lblid.Text + "' and Student\_name='" + lblname.Text + "' and Student\_Class='" + lblclass.Text + "'";

cmd.ExecuteNonQuery();

cmd.CommandText = "insert into ScholarshipEntries([Student\_id],[StudentName],[ScholarshipNumber],[Status],[StatusReason],[DateTime]) values('" + lblid.Text + "','" + lblname.Text + "','" + txtregno.Text + "','" + ddlstatus.Text + "','" +ddlDispenseStatus.Text+ "','" + DateTime.Now.ToString() + "')";

cmd.ExecuteNonQuery();

cmd.CommandText = "insert into ScholarshipUpdateData([Student\_id],[ScholarshipRegNo],[DateOfRegister],[ApplicationStatus],[ScholarshipStatus],[AmountRecievedDate],[AmountReceived],[DateOfScholarshipPaid]) values('" + lblid.Text + "','" + txtregno.Text + "','" + txtregdate.Text + "','" + ddlstatus.Text + "','" + ddlDispenseStatus.Text + "','"+txtamtRecDate.Text+"','" + txtamtreceived.Text + "','" + txtpaiddate.Text + "')";

cmd.ExecuteNonQuery();

SqlDataReader dr = cmd.ExecuteReader();

lblWhatHappened.Text = "Transaction Success";

dr.Close();

con.Close();

}

}

catch (SqlException exe)

{

lblWhatHappened.ForeColor = System.Drawing.Color.Red;

lblWhatHappened.Text = "Transaction Failed!";

Response.Write("Message From Manjunath:" + exe);

}

}

11.Student Fee Pay C# Code

namespace VirtualCampus

{

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

{

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

{

cmdpayfee.Enabled = false;

}

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

{

txtpayingfee.Text = "";

txtRpayingfee.Text = "";

}

protected void ListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

txtpayingfee.Text = "";

txtRpayingfee.Text = "";

string jesus = ListBox1.SelectedItem.Value;

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

string sql;

sql = "select \* from StudentRegistrations where Student\_id ='" + jesus + "'";

cmd.CommandText = sql;

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblWhatHappened.Text = "Student record Found!";

lblID.Text = dr["Student\_id"].ToString();

lblName.Text = dr["FirstName"].ToString();

lblClass.Text = dr["Course"].ToString();

lblSection.Text = dr["Semester"].ToString();

lblRollNo.Text = dr["RollNo"].ToString();

txttotfee.Text = dr["TotalFee"].ToString();

txtpaidfee.Text = dr["PayingFee"].ToString();

txtremfee.Text = dr["RemainingFee"].ToString();

}

else

{

dr.Close();

lblID.Text = "";

lblName.Text = "";

lblClass.Text = "";

lblSection.Text = "";

lblRollNo.Text = "";

lblWhatHappened.Text = "Student Record Doesn't Exist!";

}

}

catch (IndexOutOfRangeException ex)

{

ListBox1.ClearSelection();

Response.Write("Message From Manjunath:" + ex);

}

catch (SqlException sql)

{

Response.Write("Message From Manjunath:" + sql);

}

}

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

{

if (txtpayingfee.Text != txtRpayingfee.Text)

{

string display = "Paying Fee & Re-Enter paying Fee Doesn't Match!";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

}

else if (txtpayingfee.Text == "")

{

string display1 = "Please Enter paying Fee Inorder To Register";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display1 + "');", true);

}

else if (txtRpayingfee.Text == "")

{

string display2 = "Please Re-Enter paying Fee Inorder To Register";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display2 + "');", true);

}

else

{

try

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText= "update StudentRegistrations SET TotalFee='" + txttotfee.Text + "',RemainingFee='" + txtremfee.Text + "',PayingFee='" + txtRpayingfee.Text + "' where Student\_id='" + lblID.Text + "'";

cmd.ExecuteNonQuery();

cmd.CommandText = "insert into StudentFeeEntries([Student\_id],[Student\_Name],[Class],[Section],[RollNo],[PaidFee],[RemainingFee],[DateTime]) values('" + lblID.Text + "','" + lblName.Text + "','" + lblClass.Text + "','" + lblSection.Text + "','" + lblRollNo.Text + "','" + txtRpayingfee.Text + "','" + txtremfee.Text + "','"+DateTime.Now.ToString()+"')";

cmd.ExecuteNonQuery();

SqlDataReader dr1 = cmd.ExecuteReader();

if (dr1.Read())

{

string d = DateTime.Now.ToString();

string display = "Record Updated Failed!!please try again..thank you";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "','" + d + "');", true);

lblTransStatus.ForeColor = System.Drawing.Color.Red;

lblTransStatus.Text = "Transaction Failed!!! Try Again Later";

}

else

{

string d2 = DateTime.Now.ToString();

string display2 = "Record Update Successfully At:" + d2 + "Thank You!";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display2 + "');", true);

lblTransStatus.ForeColor = System.Drawing.Color.DarkGreen;

lblTransStatus.Text = "Transaction Successful! Thank You For Using Service";

txtpayingfee.Text = "";

txtRpayingfee.Text = "";

}

}

catch (SqlException sql)

{

lblTransStatus.Text = "Service Stopped!! Server Down...Sorry Regretted";

}

catch (FormatException frm)

{

lblTransStatus.Text = "Format Exception!!Service Stopped";

}

}

}

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

{

try

{

decimal i = decimal.Parse(txtRpayingfee.Text);

decimal j = decimal.Parse(txtremfee.Text);

if (txtpayingfee.Text.ToString() != txtRpayingfee.Text.ToString())

{

lblTransStatus.Text = "Paying Fee Doesn't Match Re-Enter Fee!";

}

else if (i>j)

{

string display2 = "Paying Fee Is Greater Than Remaining Fee! Please Check & Calculate";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display2 + "');", true);

}

else

{

txtremfee.Text = (Convert.ToDecimal(txtremfee.Text) - Convert.ToDecimal(txtpayingfee.Text)).ToString();

lblTransStatus.Text = "Calculated Successfully! You Can Update Now";

cmdpayfee.Enabled = true;

}

}

catch (FormatException format)

{

lblTransStatus.Text = "Can't Update...Student Might Not Selected or You Might Missing Remaining Fee!";

}

}

}

}

12.Files Download C# Code

namespace VirtualCampus

{

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

{

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

{

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

try

{

string str = "Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

SqlConnection con = new SqlConnection(str);

if (e.CommandName == "Download")

{

int index = Convert.ToInt32(e.CommandArgument);

GridViewRow row = GridView1.Rows[index];

string filename = row.Cells[2].Text.ToString();

string infile = Server.MapPath("~/StudentUploads/" + filename);

string outfile = @"D://" + filename;

File.Copy(infile, outfile);

lblDownDel.ForeColor = System.Drawing.Color.DarkGreen;

lblDownDel.Text = "File Downloaded to D:// Drive";

}

}

catch (IOException exe)

{

lblDownDel.Text = "File Can't Be Downloaded,File Already Downloaded & Exists In D:// Directory!";

}

}

}

}

13.File Management Download/Upload C# Code

namespace VirtualCampus

{

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

{

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

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

string sql = "select \* from StudentRegistrations where Student\_id='" + Session["NEW"] + "' ";

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblCurrentUser.Text = dr["Student\_id"].ToString();

lblCurrentUsername.Text = dr["FirstName"].ToString();

}

else

{

lblConStatus.ForeColor = System.Drawing.Color.DarkRed;

lblConStatus.Text = "Server error! Can't Start Service!";

con.Close();

}

}

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

{

lblDownDel.Text = "";

string str = "Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

SqlConnection con = new SqlConnection(str);

string cnt = "Select count(\*) from Studentfiles";

object count;

try

{

con.Open();

lblConStatus.Text = "Connection Established";

if (FileUpload1.HasFile)

{

SqlCommand cmd = new SqlCommand(cnt, con);

count = cmd.ExecuteScalar();

int id = Convert.ToInt32(lblCurrentUser.Text);

string rem = txtRemarks.Text.ToString();

int c = Convert.ToInt32(count);

string a = "Insert into Studentfiles(Student\_id,File\_id,Filename,UploadedDate,Remarks) Values(@Student\_id,@File\_id,@Filename,@Date,@Remarks)";

SqlCommand cmd2 = new SqlCommand(a, con);

cmd2.Parameters.AddWithValue("@Student\_id", id);

cmd2.Parameters.AddWithValue("@File\_id", ++c);

cmd2.Parameters.AddWithValue("@Filename", FileUpload1.FileName);

cmd2.Parameters.AddWithValue("@Remarks", rem);

cmd2.Parameters.AddWithValue("@Date",DateTime.Now.ToString());

cmd2.ExecuteNonQuery();

string filename = FileUpload1.FileName;

FileUpload1.SaveAs(Server.MapPath("~/StudentUploads/") + filename);

lbluploadstatus.Text = "File Uploaded Successfully!";

GridView1.DataBind();

}

}

catch (Exception ex)

{

lbluploadstatus.Text = "Error Uploading File To The Server!Please Retry Again Later";

}

finally

{

con.Close();

}

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

try

{

string str = "Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

SqlConnection con = new SqlConnection(str);

if (e.CommandName == "Download")

{

int index = Convert.ToInt32(e.CommandArgument);

GridViewRow row = GridView1.Rows[index];

string filename = row.Cells[2].Text.ToString();

string infile = Server.MapPath("~/StudentUploads/" + filename);

string outfile = @"D://" + filename;

File.Copy(infile, outfile);

lblDownDel.Text = "File Downloaded to D:// Drive";

}

if (e.CommandName == "Delete")

{

con.Open();

string q = "delete from Studentfiles where Filename=@Filename";

int index = Convert.ToInt32(e.CommandArgument);

GridViewRow row = GridView1.Rows[index];

string filename = row.Cells[1].Text.ToString();

SqlCommand cmd = new SqlCommand(q, con);

cmd.Parameters.AddWithValue("@Filename", filename);

cmd.ExecuteNonQuery();

string infile = Server.MapPath("~/StudentUploads/" + filename);

File.Delete(infile);

lblDownDel.Text = "File Deleted By User Command";

}

}

catch (IOException exe)

{

lblDownDel.Text = "File Can't Be Downloaded,File Already Downloaded & Exists In D:// Directory!";

}

}

}

}

14. Student Registration C# Code

namespace VirtualCampus

{

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

{

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

{

txtpayingfee.Attributes.Add("onkeypress","return NumberOnly()");

txtDOA.Text = DateTime.Now.ToString();

txttotfee.Text = "25000";

if (!IsPostBack)

{

List<string> ls = new List<string>();

ls.Add("--Select--");

ls.Add("India");

ls.Add("USA");

ls.Add("UK");

ls.Add("Japan");

ls.Add("Nigeria");

ls.Add("Africa");

DDLCountry.DataSource = ls;

DDLCountry.DataBind();

List<string> list = new List<string>();

list.Add("--Select--");

list.Add("Hindu");

list.Add("Muslim");

list.Add("Christian");

list.Add("Sikh");

list.Add("Buddhism");

list.Add("Anglo Indian");

DDLReligion.DataSource = list;

DDLReligion.DataBind();

List<string> lsCourse = new List<string>();

lsCourse.Add("--Select--");

lsCourse.Add("BCA");

lsCourse.Add("BBM");

lsCourse.Add("B.Com");

lsCourse.Add("B.A");

DDLCourse.DataSource = lsCourse;

DDLCourse.DataBind();

}

}

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

{

string dt1 = "" + DDLDay.SelectedValue.ToString() + "/" + DDLMonth.SelectedValue.ToString() + "/" + DDLYear.SelectedValue.ToString() + "";

DateTime dt = Convert.ToDateTime(dt1);

string country = DDLCountry.SelectedValue.ToString();

string state = DDLState.SelectedValue.ToString();

string city = DDLCity.SelectedValue.ToString();

string Relationship = DDLRelationship.SelectedValue.ToString();

string Occupation = DDLOccupation.SelectedValue.ToString();

string Dependents = DDLDependents.SelectedValue.ToString();

string Religion = DDLReligion.SelectedValue.ToString();

string Category = DDLCategory.SelectedValue.ToString();

string Caste = DDLCaste.SelectedValue.ToString();

string MotherTongue = DDLMotherTongue.SelectedValue.ToString();

string promo = DDLPromo.SelectedValue.ToString();

string Course = DDLCourse.SelectedValue.ToString();

string Sem = DDLSemester.SelectedValue.ToString();

string Section = DDLSection.SelectedValue.ToString();

string Gender = "";

if (RadioButton1.Checked)

{

Gender = RadioButton1.Text;

}

else if (RadioButton2.Checked)

{

Gender = RadioButton2.Text;

}

else if (RadioButton3.Checked)

{

Gender = RadioButton3.Text;

}

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Serverdatabase.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "insert into StudentRegistrations([FirstName],[MiddleName],[LastName],[MotherName],[FatherName],[GuardianName],[Gender],[BirthPlace],[DOB],[Nationality],[Age],[BloodGroup],[Country],[State],[City],[PinCode],[ContactNumber],[MobileNumber],[E-Mail],[LinkedINID],[FacebookID],[TwitterID],[NaukriID],[BlogAddress],[Relationship],[Occupation],[AnnualIncome],[Dependents],[PhoneHome],[Religion],[Category],[Caste],[MotherTongue],[CollegeName],[CollegeAddress],[PCourse],[PYearPassed],[Promo],[DateOfAdmission],[Course],[Semester],[Section],[RollNo],[TotalFee],[RemainingFee],[PayingFee],[PermanentAddress],[Username],[Password]) values('" + txtfname.Text + "','" + txtmname.Text + "','" + txtlname.Text + "','" + txtmtname.Text + "','" + txtftname.Text + "','" + txtgname.Text + "','" + Gender + "','" + txtbirthPlace.Text + "','" + dt + "','" + txtNationality.Text + "','" + txtAge.Text + "','" + txtBloodGroup.Text + "','" + country + "','" + state + "','" + city + "','" + txtpin.Text + "','" + txtContact.Text + "','" + txtMobile.Text + "','" + txtMail.Text + "','" + txtfbID.Text + "','" + txtLinkedIN.Text + "','" + txtTwitterID.Text + "','" + txtNaukriID.Text + "','" + txtBlogAddress.Text + "','" + Relationship + "','" + Occupation + "','" + DDLAnnualIncome.SelectedValue + "','" + Dependents + "','" + txtPhoneHome.Text + "','" + Religion + "','" + Category + "','" + Caste + "','" + MotherTongue + "','" + txtCollegeName.Text + "','" + txtAddress.Text + "','" + txtPCourse.Text + "','" + txtPYear.Text + "','" + promo + "','" + DateTime.Now.ToString() + "','" + Course + "','" + Sem + "','" + Section + "','" + txtrollno.Text + "','" + txttotfee.Text + "','" + txtremfee.Text + "','" + txtpayingfee.Text + "','" + txtAddress.Text + "','" + txtUsername.Text + "','" + txtPassword.Text + "')";

cmd.ExecuteNonQuery();

cmd.CommandText = "select \* from StudentRegistrations where FirstName ='" + txtfname.Text + "'";

cmd.ExecuteNonQuery();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

lblRegID.Text = dr["Student\_id"].ToString();

string d = DateTime.Now.ToString();

string display = "Congratulations! Student Registered At : " + d +"Student ID is :"+lblRegID.Text+ "Status :"+lblRegistration.Text;

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

}

dr.Close();

if (DDLCategory.SelectedValue != "General")

{

cmd.CommandText = "insert into ScholarshipStat([Student\_id],[Student\_name],[Student\_Class],[Student\_Semester],[Student\_Caste],[Student\_SubCaste]) values('" + lblRegID.Text + "','" + txtfname.Text + "','" + Course + "','" + DDLSemester.SelectedValue + "','" + Category + "','" + Caste + "')";

cmd.ExecuteNonQuery();

cmd.CommandText = "insert into StudentUserAccounts([Student\_id],[Username],[Password]) values('" + lblRegID.Text + "','" + txtUsername.Text + "','" + txtPassword.Text + "')";

cmd.ExecuteNonQuery();

lblRegistration.Text = "Registered Successfully(Scholarship Eligible)";

string d = DateTime.Now.ToString();

string display = "Congratulations! Student Registered At : " + d + "Student ID is :" + lblRegID.Text + "Status :" + lblRegistration.Text;

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

txtfname.Text = "";

txtmname.Text = "";

txtlname.Text = "";

RadioButton1.Checked = false;

RadioButton2.Checked = false;

RadioButton3.Checked = false;

txtbirthPlace.Text = "";

txtmtname.Text = "";

txtftname.Text = "";

txtgname.Text = "";

DDLDay.SelectedValue = "";

DDLMonth.SelectedValue = "";

DDLYear.SelectedValue = "";

txtNationality.Text = "";

txtAge.Text = "";

txtBloodGroup.Text = "";

city = "";

txtpin.Text = "";

txtContact.Text = "";

txtMobile.Text = "";

txtMail.Text = "";

txtLinkedIN.Text = "";

txtNaukriID.Text = "";

txtBlogAddress.Text = "";

txtTwitterID.Text = "";

txtfbID.Text = "";

Relationship = "";

Occupation = "";

DDLAnnualIncome.SelectedValue = "";

DDLDependents.SelectedValue = "";

txtPhoneHome.Text = "";

Religion = "";

Category = "";

Caste = "";

MotherTongue = "";

txtCollegeName.Text = "";

txtPAddress.Text = "";

txtPCourse.Text = "";

txtPYear.Text = "";

promo = "";

txtDOA.Text = "";

Course = "";

Sem = "";

Section = "";

txtrollno.Text = "";

txttotfee.Text = "";

txtremfee.Text = "";

txtpayingfee.Text = "";

txtAddress.Text = "";

txtUsername.Text = "";

txtPassword.Text = "";

}

else

lblRegistration.Text = "Registered Successfully(Non-Scholarship)";

con.Close();

txtfname.Text = "";

txtmname.Text = "";

txtlname.Text = "";

RadioButton1.Checked = false;

RadioButton2.Checked = false;

RadioButton3.Checked = false;

txtbirthPlace.Text = "";

txtmtname.Text = "";

txtftname.Text = "";

txtgname.Text = "";

DDLDay.SelectedValue = "";

DDLMonth.SelectedValue = "";

DDLYear.SelectedValue = "";

txtNationality.Text = "";

txtAge.Text = "";

txtBloodGroup.Text = "";

txtpin.Text = "";

txtContact.Text = "";

txtMobile.Text = "";

txtMail.Text = "";

txtLinkedIN.Text = "";

txtNaukriID.Text = "";

txtBlogAddress.Text = "";

txtTwitterID.Text = "";

txtfbID.Text = "";

Relationship = "";

Occupation = "";

DDLAnnualIncome.SelectedValue = "";

Dependents = "";

txtPhoneHome.Text = "";

Religion = "";

Category = "";

Caste = "";

MotherTongue = "";

txtCollegeName.Text = "";

txtPAddress.Text = "";

txtPCourse.Text = "";

txtPYear.Text = "";

promo = "";

txtDOA.Text = "";

Course = "";

Sem = "";

Section = "";

txtrollno.Text = "";

txttotfee.Text = "";

txtremfee.Text = "";

txtpayingfee.Text = "";

txtAddress.Text = "";

txtUsername.Text = "";

txtPassword.Text = "";

string d1 = DateTime.Now.ToString();

string display1 = "Congratulations! Student Registered At : " + d1 + "Student ID is :" + lblRegID.Text + "Status :" + lblRegistration.Text;

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display1 + "');", true);

}

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

{

try

{

if (txtpayingfee.Text == "")

{

string display = "Please Enter paying Fee Inorder To Register";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

}

else

{

txtremfee.Text = (Convert.ToDecimal(txttotfee.Text) - Convert.ToDecimal(txtpayingfee.Text)).ToString();

}

}

catch (FormatException fex)

{

string display = "Fee Must Be In Indian Rupees Or Numbers!Other Characters Are Not Allowed";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + display + "');", true);

}

}

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

{

txtfname.Text = "";

}

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

{

Response.Redirect("interface.aspx");

}

protected void cmdFeecalc\_Click1(object sender, EventArgs e)

{

txtremfee.Text = (Convert.ToDecimal(txttotfee.Text) - Convert.ToDecimal(txtpayingfee.Text)).ToString();

}

protected void DDLCountry\_SelectedIndexChanged(object sender, EventArgs e)

{

List<string> ls = new List<string>();

ls.Clear();

ls.Add("Select State");

switch (DDLCountry.Text)

{

case "India":

ls.Add("Karnataka");

ls.Add("Tamilnadu");

ls.Add("Kerala");

ls.Add("Andhra Pradesh");

break;

case "US":

ls.Add("Washington");

ls.Add("England");

break;

}

DDLState.DataSource = ls;

DDLState.DataBind();

}

protected void DDLState\_SelectedIndexChanged(object sender, EventArgs e)

{

List<string> ls = new List<string>();

ls.Clear();

ls.Add("Select City");

switch (DDLState.Text)

{

case "TamilNadu":

ls.Add("Chennai");

ls.Add("Vellore");

ls.Add("Hosur");

break;

case "Karnataka":

ls.Add("Bangalore");

ls.Add("Mysore");

ls.Add("Glbarga");

ls.Add("Chitradurga");

ls.Add("Bidar");

ls.Add("Shimoga");

ls.Add("Hubli");

ls.Add("Chikkamagaluru");

ls.Add("Hassan");

break;

case "Washington":

ls.Add("Washington City");

ls.Add("Wash 2");

break;

case "England":

ls.Add("Eng 1");

ls.Add("Eng 2");

break;

}

DDLCity.DataSource = ls;

DDLCity.DataBind();

}

protected void DDLReligion\_SelectedIndexChanged(object sender, EventArgs e)

{

List<string> ls = new List<string>();

ls.Clear();

ls.Add("--Select--");

switch (DDLReligion.Text)

{

case "Hindu":

ls.Add("SC");

ls.Add("ST");

ls.Add("OBC");

ls.Add("General");

break;

case "Muslim":

ls.Add("Muslim");

break;

case "Christian":

ls.Add("Roman");

ls.Add("Catholic");

break;

case "Sikh":

ls.Add("Sikh");

break;

case "Anglo Indian":

ls.Add("Foreign Indian");

break;

}

DDLCategory.DataSource = ls;

DDLCategory.DataBind();

}

protected void DDLCategory\_SelectedIndexChanged(object sender, EventArgs e)

{

List<string> ls = new List<string>();

ls.Clear();

ls.Add("--Select--");

switch (DDLCategory.Text)

{

case "SC":

ls.Add("Adi Karnataka");

ls.Add("Adi Dravida");

break;

case "ST":

ls.Add("Kuruba");

break;

case "OBC":

ls.Add("Vokkaliga");

break;

case "General":

ls.Add("Gowda's");

ls.Add("Reddy's");

ls.Add("lingaaytath's");

ls.Add("Aachaari");

break;

case "Muslim":

ls.Add("Muslim 1");

break;

case "Roman":

ls.Add("Rome 1");

break;

case "Catholic":

ls.Add("Catholic 1");

break;

case "Sikh":

ls.Add("Sikh 1");

break;

case "Buddhism":

ls.Add("Buddhism 1");

break;

case "Foreign Indian":

ls.Add("Anglo Indian Empty");

break;

}

DDLCaste.DataSource = ls;

DDLCaste.DataBind();

}

protected void DDLCourse\_SelectedIndexChanged(object sender, EventArgs e)

{

List<string> lsC = new List<string>();

lsC.Clear();

lsC.Add("--Select--");

switch (DDLCourse.Text)

{

case "BCA":

lsC.Add("--Select--");

lsC.Add("BCA 1st Year");

lsC.Add("BCA 2nd Year");

lsC.Add("Bca 3rd Year");

break;

case "BBM":

lsC.Add("--Select--");

lsC.Add("BBM 1st Year");

lsC.Add("BBM 2nd Year");

lsC.Add("BBM 3rd Year");

break;

case "B.A":

lsC.Add("BA 1st Year");

lsC.Add("BA 2nd Year");

lsC.Add("BA 3rd Year");

break;

case "B.Com":

lsC.Add("BCOM 1st YearS");

lsC.Add("BCOM 2nd Year");

lsC.Add("BCOM 3rd Year");

break;

}

DDLSemester.DataSource = lsC;

DDLSemester.DataBind();

}

private void OnTextChanged(object sender, EventArgs e)

{

if (!System.Text.RegularExpressions.Regex.IsMatch("^[0-9]", txtAge.Text))

{

txtAge.Text = string.Empty;

}

}

protected void txtAge\_TextChanged(object sender, EventArgs e)

{

if (!System.Text.RegularExpressions.Regex.IsMatch("^[0-9]", txtAge.Text))

{

txtAge.Text = string.Empty;

}

}

}

}

TESTING

TEST CASE DESIGNS

**TESTING PLAN**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

**STRATEGIC APPROACH TO SOFTWARE TESTING**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

**CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**HYBRID TESTING**.

Hybrid testing examines class operation at an algorithmic granularity but only examines public methods and variables. From the standard point of the application, hybrid testing qualifies as white box testing since the classes being tested may not be exposed to the application. From the class level, hybrid testing qualifies as black box testing since private methods and variables are not exposed and how the results are produced is never called into question.

**Test Reports & Analysis**

**COMPILATION TEST:-**

It was a good idea to do our stress testing early on, because it gives us time to fix some of unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

**EXECUTION TEST:-**

This Program was successfully loaded and executed. Because of good programming there were no execution errors

**OUTPUT TEST:-**

The Successful output screens are placed in the output screen section**.**

INPUT-OUTPUT SCREENS

INPUT-OUTPUT SCREENS

CONCLUSION

CONCLUSION

FUTURE ENHANCEMENTS

FUTURE ENHANCEMENTS

* SMS Notification Service For Login’s And Logout’s.
* Adding Courses Manager.
* Adding Attendance Manger.
* Adding Time Table Manager.
* Third party Scholarship Registration On The Go.
* E-Mail Notifications.
* Automatic Attendance Calculator.
* Exam Hall Ticket Delivery System.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Google Now
2. Microsoft MSDN
3. [www.stackoverflow.com](http://www.stackoverflow.com)
4. W3 Schools
5. Asp.Net Forum