ACKNOWLEDGEMENT

The successful presentation of the **DBMS MINI PROJECT** would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the DBMS Mini Project.

I would like to extend my gratitude to **Dr. K. RAMA NARASIMHA,** Principal / Director, K. S. School of Engineering and Management, Bengaluru, for facilitating me to present the DBMS Mini Project.

I thank **Dr. VANDANA JHA.**, Associate Professor and Head, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for her encouragement.

I would like to thank our Project Guide, **Dr. PRADEEP.K. R**, Assistant Professor, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for their constant guidance and inputs.

I would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their co-operation.

Finally, I extend my heart-felt gratitude to my family for their encouragement and support without which I wouldn’t have come so far. Moreover, I thank all my friends for their invaluable support and cooperation.

NANDEESH S

VENKATESH

ABSTRACT

As most of the work is done manually or it is based on paper work such as class attendance, notes, library dues, asking for documents etc. These all process takes time. If we include all the work which will be based on online system, then it can reduce time and work.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 OVERVIEW

A college management system enables colleges to efficiently retrieve, add, update, delete student data such as name, address, marks etc

It is outdated and difficult to maintain such a huge information manually.

Thankfully we have dbms and with the use of dbms we have created this software which helps the admin to maintain the data about students faster and efficiently

1.2 PROBLEM STATEMENT

The main aim and objective of this project is to reduce the manual work of the administrator by helping in maintaining all the records computerised which is safe and secure.

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database’s logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13]Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 MICROSOFT VISUAL STUDIO

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer)

1.6 **INTRODUCTION TO C#**

# is a general-purpose, modern and object-oriented programming language C pronounced as **"C sharp"**. It was developed by Microsoft led by Anders Hejlsberg and his team within the .Net initiative and was approved by the European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is among the languages for Common Language Infrastructure and the current version of C# is version 7.2. C# is a lot similar to Java syntactically and is easy for the users who have knowledge of C, C++ or Java

You can use Visual Studio to create and update a local database file in SQL Server Express LocalDB. You can also create a database by executing Transact-SQL statements in the SQL Server Object Explorer tool window in Visual Studio. In this topic, we'll create an *.mdf* file and add tables and keys by using the Table Designer.

1.7 CREATE A PROJECT AND ADD LOCAL DATABASE FILE USING VISUAL STUDIO

1. Create a new Windows Forms App (.NET Framework) project and name it

2. Sample Database Walkthrough.

3. On the menu bar, select Project > Add New Item.

4. In the list of item templates, scroll down and select Service-based Database.

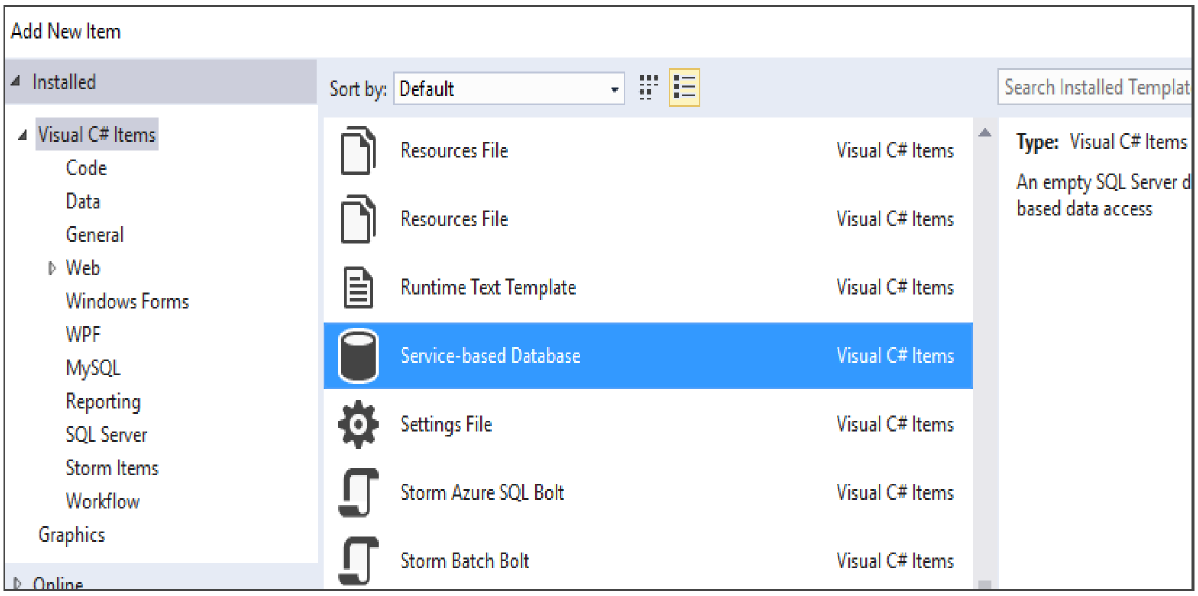

Add New Item
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
MySQL
Reporting
SQL Server
Storm Items
Workflow
Graphics
Online
Sort by: Default
Search Installed Templat
Resources File — Visual C# Items
Resources File — Visual C# Items
Runtime Text Template — Visual C# Items
Service-based Database — Visual C# Items
Settings File — Visual C# Items
Storm Azure SQL Bolt — Visual C# Items
Storm Batch Bolt — Visual C# Items
Type: Visual C# Items
An empty SQL Server d
based data access

Fig 1.1: adding database

5.Name the database Sample Database, and then click Add.

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of college management system are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

- ◻ software - visual studio 2019
- ◻ version - community
- ◻ operating system - windows10

2.2.2 HARDWARE REQUIREMENTS

- ◻ Processor –i5 or above
- ◻ RAM – 4GB or above
- ◻ Hard disk – 10 GB or above
- ◻ Monitor – VGA of 1024x768 screen resolution
- ◻ Keyboard and Mouse

CHAPTER 3

DETAILED DESIGN

3.1 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.

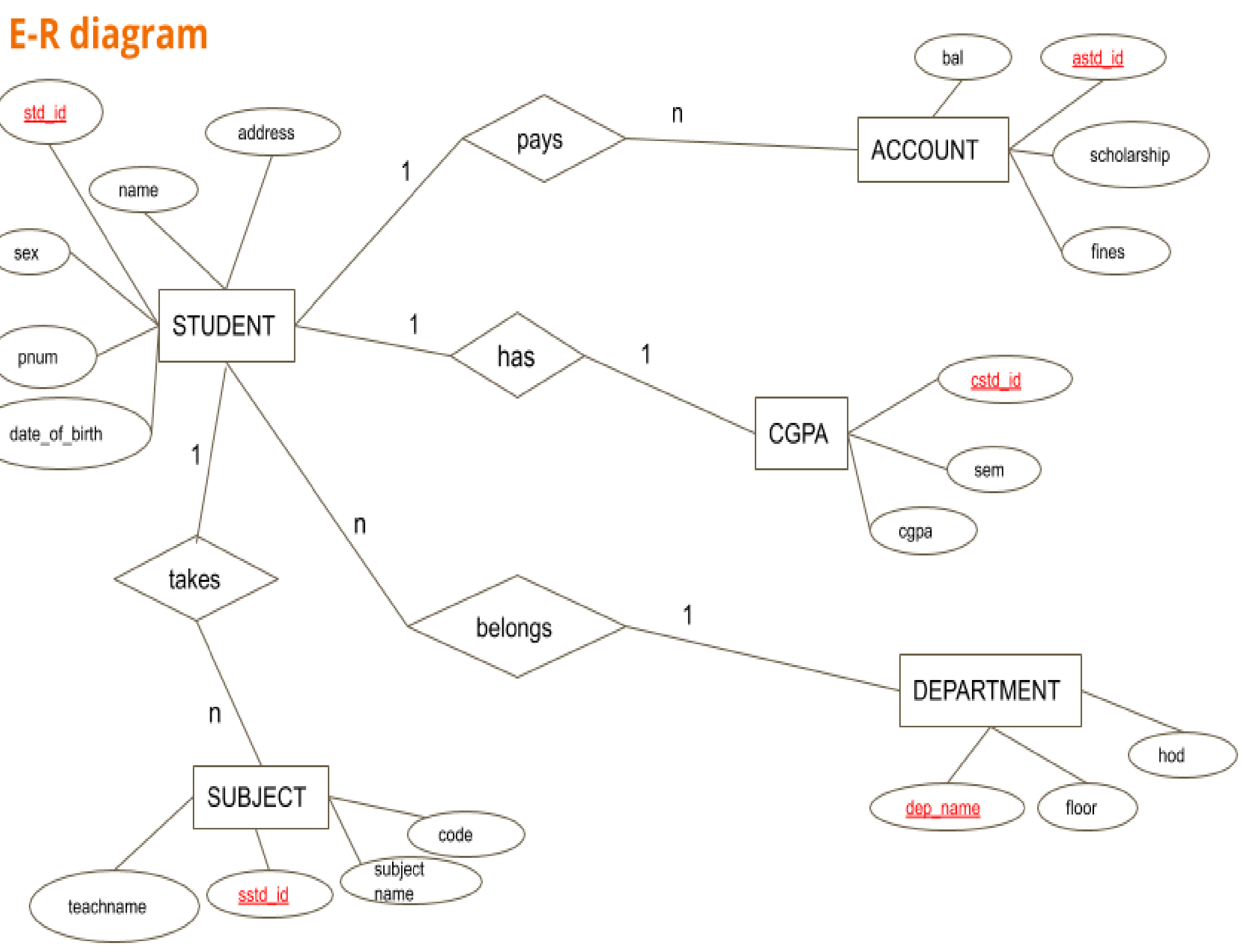


Fig 3.1 ER Diagram

`

3.2 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.

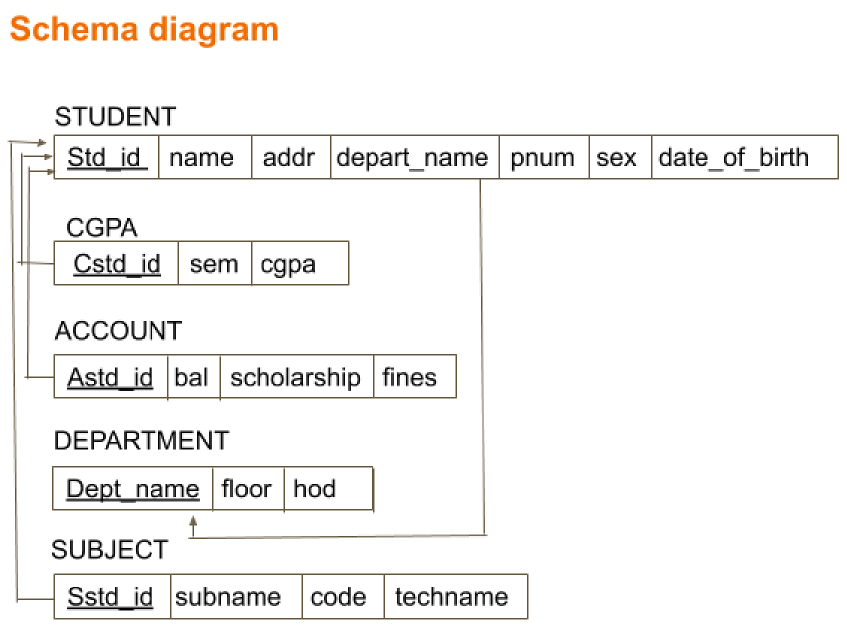


Fig 3.2 Schema diagram

3.3 DESCRIPTION OF TABLES

The database consists of six tables:

1. STUDENT: It stores the general student details
- ◻ std_id: Unique student id done by auto increment.
- ◻ Name: Name of the student.
- ◻ addr: address of the user
- ◻ depart_name:the forigen key of student table associated with depart_name
- ◻ Pnum: Phone number of the user.
- ◻ sex:male or female
- ◻ date_of_birth:the birth date

2. CGPA: It stores marks related info about student
- ◻ cstd_id: Foreign key of student associated with cgpa
- ◻ sem:sem of the student
- ◻ cgpa:cgpa of the student

3. ACCOUNT: Stores the info about the students account details
- ◻ Astd_id:foreign key of account associated with the student
- ◻ bal:balance of the student fees
- ◻ scholarship:the amount of student scholarship
- ◻ fines:the amount of fine needs to be paid by the student

4. DEPARTMENT: It stores the info about department
- ◻ dept_name:the names of the department
- ◻ floor:the floor no of the corresponding department
- ◻ hod:the name of the hod of the corresponding department

5. SUBJECT:It stores info about subject
- ◻ Sstd_id:forigen key of the subject associated with the student
- ◻ subname:the name of the subject
- ◻ code:code of the subject
- ◻ teachname:the name of the teacher

Chapter 4

IMPLEMENTATION

4.1 MODULES AND THEIR ROLES

4.1.1Login: Login for the admin

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;



namespace Nandeesh
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

            SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\repos1\\Nandeesh\\Nandeesh\\Database1.mdf;Integrated Security=True");
        SqlCommand cmd;
        SqlDataReader dr;



        private string getusername()
        {
            //fetch data from data base
            con.Open();
            string syntax = "select value from SYSTEMTABLE where property='username'";
            cmd = new SqlCommand(syntax, con);
            dr = cmd.ExecuteReader();
            dr.Read();
            string temp = dr[0].ToString();
            con.Close();
            return temp;
        }


        private string getpassword()
        {
            //fetch data from data base
            con.Open();
            string syntax = "select value from SYSTEMTABLE where property='password'";
            cmd = new SqlCommand(syntax, con);
```

```
    dr = cmd.ExecuteReader();
    dr.Read();
    string temp = dr[0].ToString();
    con.Close();
    return temp;

 }

private void button1_Click(object sender, EventArgs e)
 {

    string auname = getusername(), apass = getpassword(), euname, epass;
    euname = textBox1.Text;
    epass = textBox2.Text;

    if(auname==euname&&epass==apass)
    {
       //login
            appbody obj = new appbody();
       this.Hide();
       obj.Show();


    }
    else
    {
       //dontlogin
       MessageBox.Show("user name and password doesnt match");
    }
 }
```

```
        private void label2_Click(object sender, EventArgs e)
        {
        }
        private void Form1_Load(object sender, EventArgs e)

    }
    }}
    }
```

4.1.2homepage:students homepage

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace Nandeesh
{
    public partial class appbody : Form
    {
        public const int WM_NCLBUTTONDOWN = 0xA1;
        public const int HT_CAPTION = 0x2;

        [DllImportAttribute("user32.dll")]
        public static extern int SendMessage(IntPtr hWnd,
                 int Msg, int wParam, int lParam);
```

```
[DllImportAttribute("user32.dll")]
public static extern bool ReleaseCapture();






public appbody()
{
    InitializeComponent();
    //initialization for sliding panel
    isslidingpanelexpanded = true;
    expandslidingpanelGUI();

}

private void appbody_Load(object sender, EventArgs e)
{
    this.Opacity = 0.1;
    login_timer.Start();
}

private void login_timer_Tick(object sender, EventArgs e)
{
    if(this.Opacity<=1.0)
    {
        this.Opacity += 0.025;

    }
    else
    {
        login_timer.Stop();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}

private void close_button_Click(object sender, EventArgs e)
{
    Environment.Exit(0);
}

private void button1_Click(object sender, EventArgs e)
{
    Form1 obj = new Form1();
    this.Hide();
    obj.Show();
}

private void panel1_MouseDown(object sender, MouseEventArgs e)
{
   if(e.Button==MouseButtons.Left)
    {
        ReleaseCapture();
        SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
    }
```

```
}




//sliding panel starts from here
bool isslidingpanelexpanded;
const int maxsliderwidth=130;
const int minsliderwidth=60;


private void button1_Click_1(object sender, EventArgs e)
{
    if (isslidingpanelexpanded)
    {
        //retract panel
       //retractslidingpanelGUI();

    }
    slidingpanel_timer.Start();
}

private void slidingpanel_timer_Tick(object sender, EventArgs e)
{
    if (isslidingpanelexpanded)
    {
        //retract panel
        sliding_panel.Width -= 30;
        if (sliding_panel.Width <= minsliderwidth)
        {
            //stopretract
            isslidingpanelexpanded = false;
            slidingpanel_timer.Stop();
            retractslidingpanelGUI();
            this.Refresh();
        }
    }
    else
    {
        //expand the panel
        sliding_panel.Width += 30;
        if (sliding_panel.Width >= maxsliderwidth)
        {
            //stop expand
            isslidingpanelexpanded = true;
            slidingpanel_timer.Stop();
            expandslidingpanelGUI();
            this.Refresh();
        }
    }
}

private void sliding_panel_Paint(object sender, PaintEventArgs e)
{


}

public void expandslidingpanelGUI()
{
```

```
    //gui adjustments for expanding
    student_button.Text = "STUDENT";
    grade_button.Text = "GRADE";
    account_button.Text = "ACCOUNT";
    settings_button.Text = "SETTINGS";
    about_button.Text = "ABOUT";

    sliding_button.Image = Properties.Resources.larrow;

    student_button.Image = Properties.Resources.student;
    grade_button.Image = Properties.Resources.marks2;
    account_button.Image = Properties.Resources.acc1;
    settings_button.Image = Properties.Resources.set;
    about_button.Image = Properties.Resources.about__1_;
}
public void retractslidingpanelGUI()
{
    //gui adjustments for expanding
    student_button.Text = "";
    grade_button.Text = "";
    account_button.Text = "";
    settings_button.Text = "";
    about_button.Text = "";

    sliding_button.Image = Properties.Resources.double_chevron2;

    student_button.Image = Properties.Resources.student;
    grade_button.Image = Properties.Resources.marks2;
    account_button.Image = Properties.Resources.acc1;
    settings_button.Image = Properties.Resources.set;
    about_button.Image = Properties.Resources.about__1_;
}

private void student_button_Click(object sender, EventArgs e)
{
    if(!contentpanel.Controls.Contains(student_usercontrol.instance))
    {
        contentpanel.Controls.Add(student_usercontrol.instance);
        student_usercontrol.instance.Dock = DockStyle.Fill;
        student_usercontrol.instance.BringToFront();
    }
    else
    {
        student_usercontrol.instance.BringToFront();
    }
}

private void grade_button_Click(object sender, EventArgs e)
{
    if (!contentpanel.Controls.Contains(cgpa_usercontrol.instance))
    {

        contentpanel.Controls.Add(cgpa_usercontrol.instance);
        cgpa_usercontrol.instance.Dock = DockStyle.Fill;
        cgpa_usercontrol.instance.BringToFront();

    }
    else
    {
        cgpa_usercontrol.instance.BringToFront();
    }
}

private void account_button_Click(object sender, EventArgs e)
{
    if (!contentpanel.Controls.Contains(account_usercontrol.instance))
    {
        contentpanel.Controls.Add(account_usercontrol.instance);
        account_usercontrol.instance.Dock = DockStyle.Fill;
        account_usercontrol.instance.BringToFront();
    }
    else
    {
        account_usercontrol.instance.BringToFront();
    }
}

private void settings_button_Click(object sender, EventArgs e)
{
    if (!contentpanel.Controls.Contains(settings_usercontrol.instance))
    {
        contentpanel.Controls.Add(settings_usercontrol.instance);
        settings_usercontrol.instance.Dock = DockStyle.Fill;
        settings_usercontrol.instance.BringToFront();
    }
    else
    {
        settings_usercontrol.instance.BringToFront();
    }
}
```

```
        private void about_button_Click(object sender, EventArgs e)
        {
            if (!contentpanel.Controls.Contains(about_usercontrol.instance))
            {
                contentpanel.Controls.Add(about_usercontrol.instance);
                about_usercontrol.instance.Dock = DockStyle.Fill;
                about_usercontrol.instance.BringToFront();
            }
            else
            {
                about_usercontrol.instance.BringToFront();
            }
        }

        private void button1_Click_2(object sender, EventArgs e)
        {

            this.WindowState= FormWindowState.Maximized;

        }

        private void button1_Click_3(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Maximized;
        }
    }

}
```

4.1.3 student user control

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Nandeesh
{
    public partial class student_usercontrol : UserControl
    {
        private static student_usercontrol _instance;
        public static student_usercontrol instance
        {
            get
            {
                if(_instance==null)
                {
                    _instance = new student_usercontrol();
                }
                return _instance;
            }
        }
        public student_usercontrol()
        {
            InitializeComponent();
        }



        public void refresh_datagridview()
        {
            try
            {
                SqlCommand cmd = new SqlCommand("showallstudentdata_SP", con);
                cmd.CommandType = CommandType.StoredProcedure;
                SqlDataAdapter DA = new SqlDataAdapter(cmd);
                DataSet DS = new DataSet();
                DA.Fill(DS);

                con.Open();
                try
                {
                    cmd.ExecuteNonQuery();
                }
                catch (Exception ex)
                {
                    MessageBox.Show("            <<<INVALID SQL OPERATION>>>: \n" + ex);
                }
                con.Close();

                dataGridView1.DataSource = DS.Tables[0];

            }
            catch(Exception ex)
            {
                MessageBox.Show("" + ex);

            }

        }
```

```
        SqlConnection con = new SqlConnection("Data 
    Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\repos1\\Nandeesh\\Nandeesh
    \\Database1.mdf;Integrated Security=True");
```

```
        private void button5_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("studentadd_sp", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.AddWithValue("@std_id",stdid_textbox.Text);
            cmd.Parameters.AddWithValue("@name", name_textbox.Text);
            cmd.Parameters.AddWithValue("@addr", address_textbox.Text);
            cmd.Parameters.AddWithValue("@phnum", phno_textbox.Text);
            cmd.Parameters.AddWithValue("@sex", sex_textbox.Text);
```

```
    cmd.Parameters.AddWithValue("@date_of_birth", dob_textbox.Text);
    cmd.Parameters.AddWithValue("@depart_name", departname_textbox.Text);

    con.Open();
    try
    {
        cmd.ExecuteNonQuery();
    }
    catch(Exception ex)
    {
        MessageBox.Show("       <<<INVALID SQL OPERATION>>>: \n" + ex);
    }
    con.Close();

    refresh_datagridview();
}

private void student_usercontrol_Load(object sender, EventArgs e)
{
    refresh_datagridview();
}

private void button4_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("studentdata_delete", con);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.AddWithValue("@std_id", stdid_textbox.Text);

        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show("       <<<INVALID SQL OPERATION>>>: \n" + ex);
        }
        con.Close();

        refresh_datagridview();
    }
    catch(Exception ex)
    {
        MessageBox.Show("" + ex);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("searchstudent_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@std_id", stdid_textbox.Text);


        SqlDataAdapter DA = new SqlDataAdapter(cmd);
        DataSet DS = new DataSet();
        DA.Fill(DS);



        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show("            <<<INVALID SQL OPERATION>>>: \n" + ex);
        }
        con.Close();

        dataGridView1.DataSource = DS.Tables[0];

    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex);

    }
}

private void button3_Click(object sender, EventArgs e)
{
    stdid_textbox.Text = null;
    name_textbox.Text = null;
    address_textbox.Text = null;
    phno_textbox.Text = null;
    sex_textbox.Text = null;
    dob_textbox.Text = null;
```

```
        }

        private void button2_Click(object sender, EventArgs e)
        {

        }
    }
}
```

4.1.4 grade user control

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Nandeesh
{
    public partial class cgpa_usercontrol : UserControl
    {
        private static cgpa_usercontrol _instance;
        public static cgpa_usercontrol instance
        {
            get
            {
                if (_instance == null)
            {
                _instance = new cgpa_usercontrol();
            }
            return _instance;
        }
    }
    public cgpa_usercontrol()
    {
        InitializeComponent();
        SqlConnection con = new SqlConnection("Data
        Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\repos1\\Nandee
        sh\\Nandeesh\\Database1.mdf;Integrated Security=True");
    public void refresh_datagridview()
    {
        try
        {
            SqlCommand cmd = new SqlCommand("showstudentcgpa_sp", con);
            cmd.CommandType = CommandType.StoredProcedure;

            SqlDataAdapter DA = new SqlDataAdapter(cmd);
            DataSet DS = new DataSet();
            DA.Fill(DS);

            con.Open();
            try
            {
                cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show("          <<<INVALID SQL OPERATION>>>: \n" + ex);
                }
                con.Close();

                dataGridView1.DataSource = DS.Tables[0];
                this.dataGridView1.Columns[0].AutoSizeMode                                                    =
DataGridViewAutoSizeColumnMode.Fill;
                this.dataGridView1.Columns[1].AutoSizeMode                                                    =
DataGridViewAutoSizeColumnMode.Fill;
                this.dataGridView1.Columns[2].AutoSizeMode                                                    =
DataGridViewAutoSizeColumnMode.Fill;
            }
            catch (Exception ex)
            {
                MessageBox.Show("" + ex);

            }

        }

        private void button5_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("addstdcgpa", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.AddWithValue("@cstd_id", cstdid_textbox.Text);
            cmd.Parameters.AddWithValue("@sem", sem_textbox.Text);
            cmd.Parameters.AddWithValue("@cgpa", cgpa_textbox.Text);

            con.Open();
            try
            {
```

```
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        MessageBox.Show("       <<<INVALID SQL OPERATION>>>: \n" + ex);
    }
    con.Close();
    refresh_datagridview();




}

private void cgpa_usercontrol_Load(object sender, EventArgs e)
{
    refresh_datagridview();
}

private void button4_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("cstdiddelete_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.AddWithValue("@cstd_id", cstdid_textbox.Text);

        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show("      <<<INVALID SQL OPERATION>>>: \n" + ex);
        }
        con.Close();

        refresh_datagridview();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    cstdid_textbox.Text = null;
    sem_textbox.Text = null;
    cgpa_textbox.Text = null;
}

private void button6_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("searchcgpa_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@cstd_id",cstdid_textbox.Text);
```

```
          SqlDataAdapter DA = new SqlDataAdapter(cmd);
          DataSet DS = new DataSet();
          DA.Fill(DS);



          con.Open();
          try
          {
            cmd.ExecuteNonQuery();
          }
          catch (Exception ex)
          {
            MessageBox.Show("         <<<INVALID SQL OPERATION>>>: \n" + ex);
          }
          con.Close();

          dataGridView1.DataSource = DS.Tables[0];

        }
        catch (Exception ex)
        {
          MessageBox.Show("" + ex);

        }
      }
    }
```

```
}
```

4.1.5 account user control

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;


namespace Nandeesh
{
    public partial class account_usercontrol : UserControl
    {
        private static account_usercontrol _instance;
        public static account_usercontrol instance
        {
            get
            {
                if (_instance == null)
                {
                    _instance = new account_usercontrol();
                }
                return _instance;
            }
        }
```

```
public account_usercontrol()
{
    InitializeComponent();
}

SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\repos1\\Nandeesh\\Nandeesh\\Database1.mdf;Integrated Security=True");
public void refresh_datagridview()
{
    try
    {
        SqlCommand cmd = new SqlCommand("showaccstddata_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter DA = new SqlDataAdapter(cmd);
        DataSet DS = new DataSet();
        DA.Fill(DS);

        con.Open();
        try
          {
            cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show("          <<<INVALID SQL OPERATION>>>: \n" + ex);
        }
        con.Close();
```

```
                dataGridView1.DataSource = DS.Tables[0];
                this.dataGridView1.Columns[0].AutoSizeMode = 
        DataGridViewAutoSizeColumnMode.Fill;
                this.dataGridView1.Columns[1].AutoSizeMode = 
        DataGridViewAutoSizeColumnMode.Fill;
                this.dataGridView1.Columns[2].AutoSizeMode = 
        DataGridViewAutoSizeColumnMode.Fill;
            }
            catch (Exception ex)
            {
                MessageBox.Show("" + ex);

            }
          }
            private void button5_Click(object sender, EventArgs e)
          {
            SqlCommand cmd = new SqlCommand("addastdid_sp", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.AddWithValue("@astd_id", astdid_textbox.Text);
            cmd.Parameters.AddWithValue("@balance", bal_textbox.Text);
            cmd.Parameters.AddWithValue("@scholarship",scholarship_textbox.Text);
            cmd.Parameters.AddWithValue("@fines",fines_textBox.Text);

            con.Open();
            try
            {
                cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
```

```
        MessageBox.Show("       <<<INVALID SQL OPERATION>>>: \n" + ex);
    }
    con.Close();
    refresh_datagridview();




}

private void account_usercontrol_Load(object sender, EventArgs e)
{
    refresh_datagridview();
}

private void button4_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("accountdelete_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.AddWithValue("@astd_id",astdid_textbox.Text);

        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
                MessageBox.Show("      <<<INVALID SQL OPERATION>>>: \n" + ex);
            }
            con.Close();

            refresh_datagridview();
        }
        catch (Exception ex)
        {
            MessageBox.Show("" + ex);
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        astdid_textbox.Text = null;
        bal_textbox.Text = null;
        scholarship_textbox.Text = null;
        fines_textBox.Text = null;
    }
}
}
```

4.1.6:**log out button**

```
private void button1_Click(object sender, EventArgs e)
                {
                            Form1 obj = new Form1();
                            this.Hide();
                           obj.Show();
                    }
```

4.2 RESULT

The resulting system is able to:

- Authenticate admin credentials during login.
- Salted encryption for security of user passwords.
- Add student data.
- Delete student data
- Update student data.
- search student data

Chapter 5

TESTING

5.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

Chapter 6

SNAPSHOTS

This chapter consists of working screenshots of the project.

6.1 LOGIN PAGE

This is the login page for the admin who uses this software

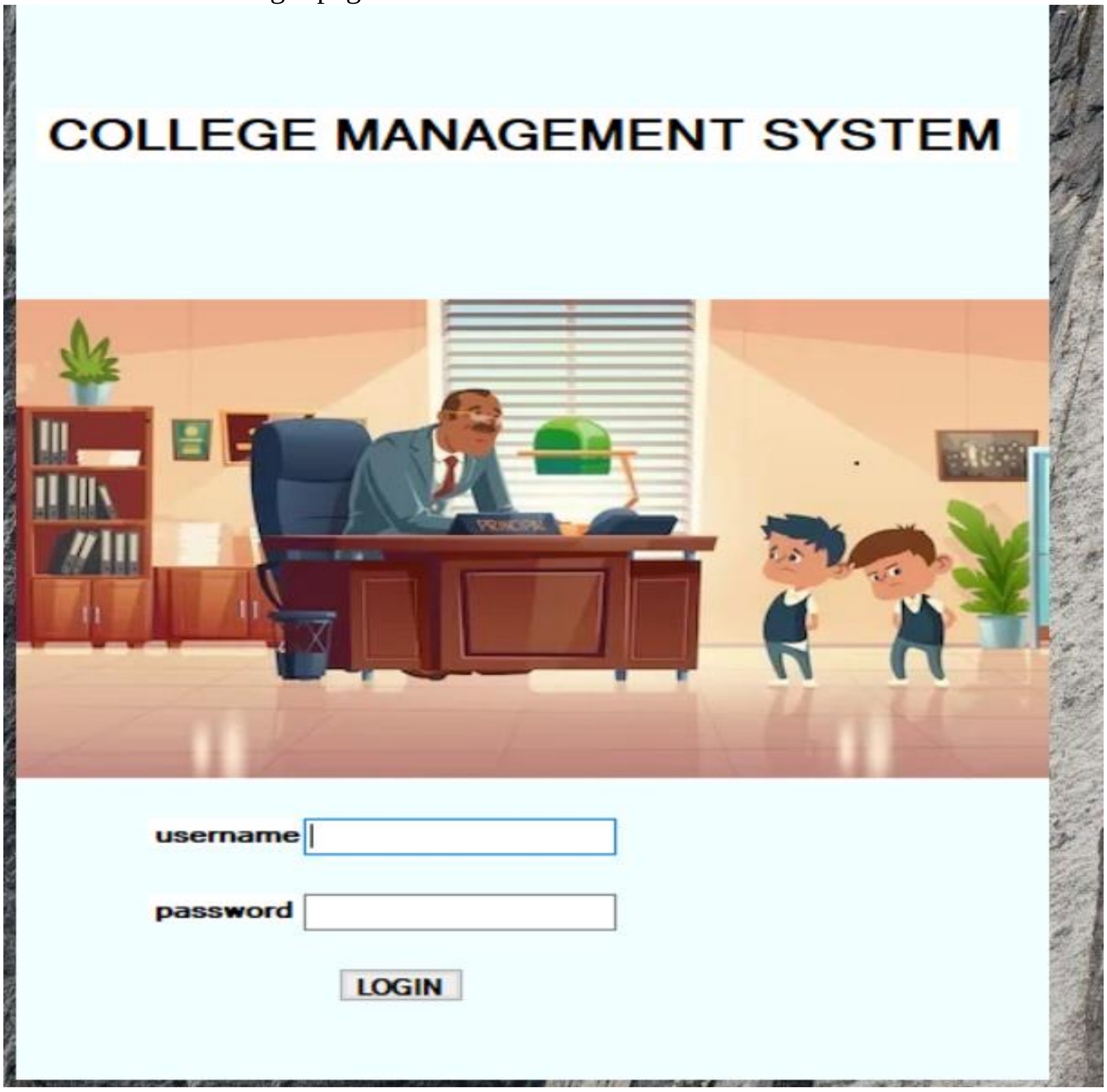


Fig 6.1: Login page

6.2 HOME PAGE

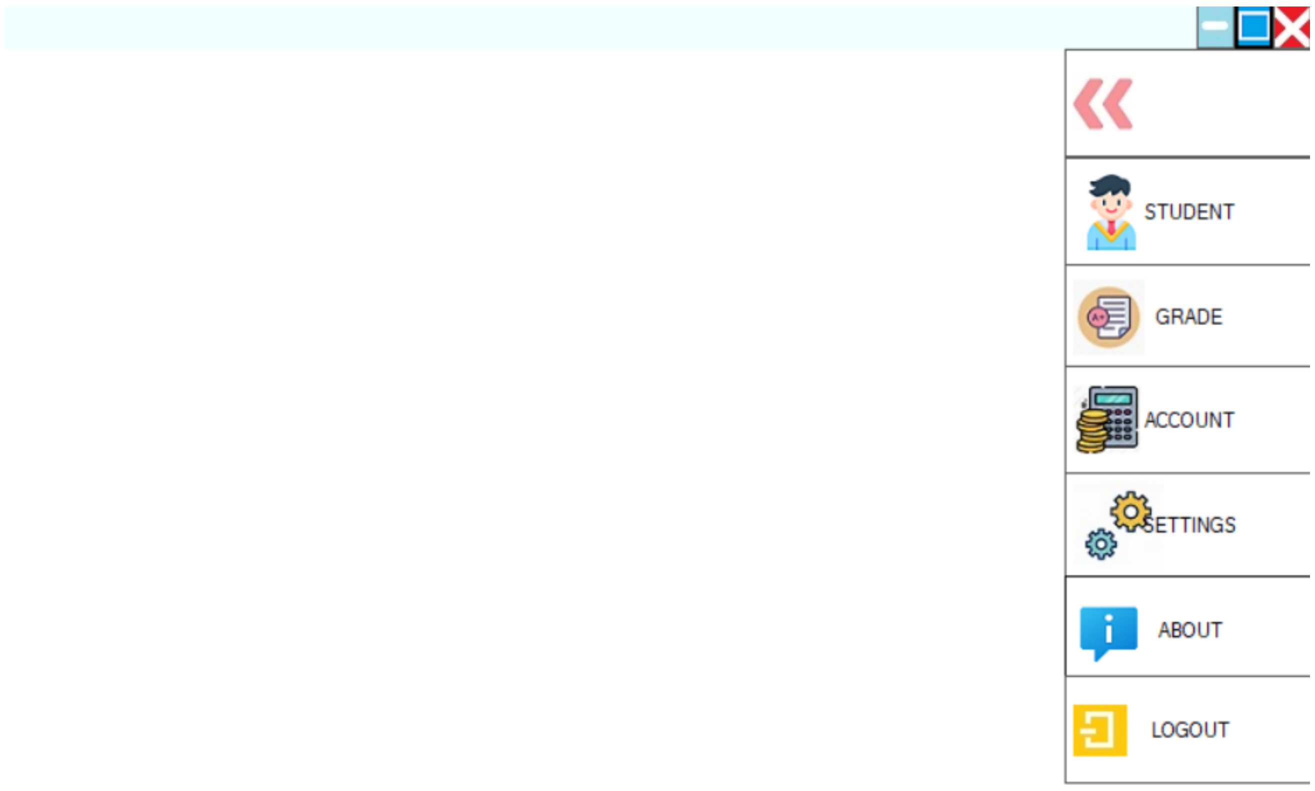
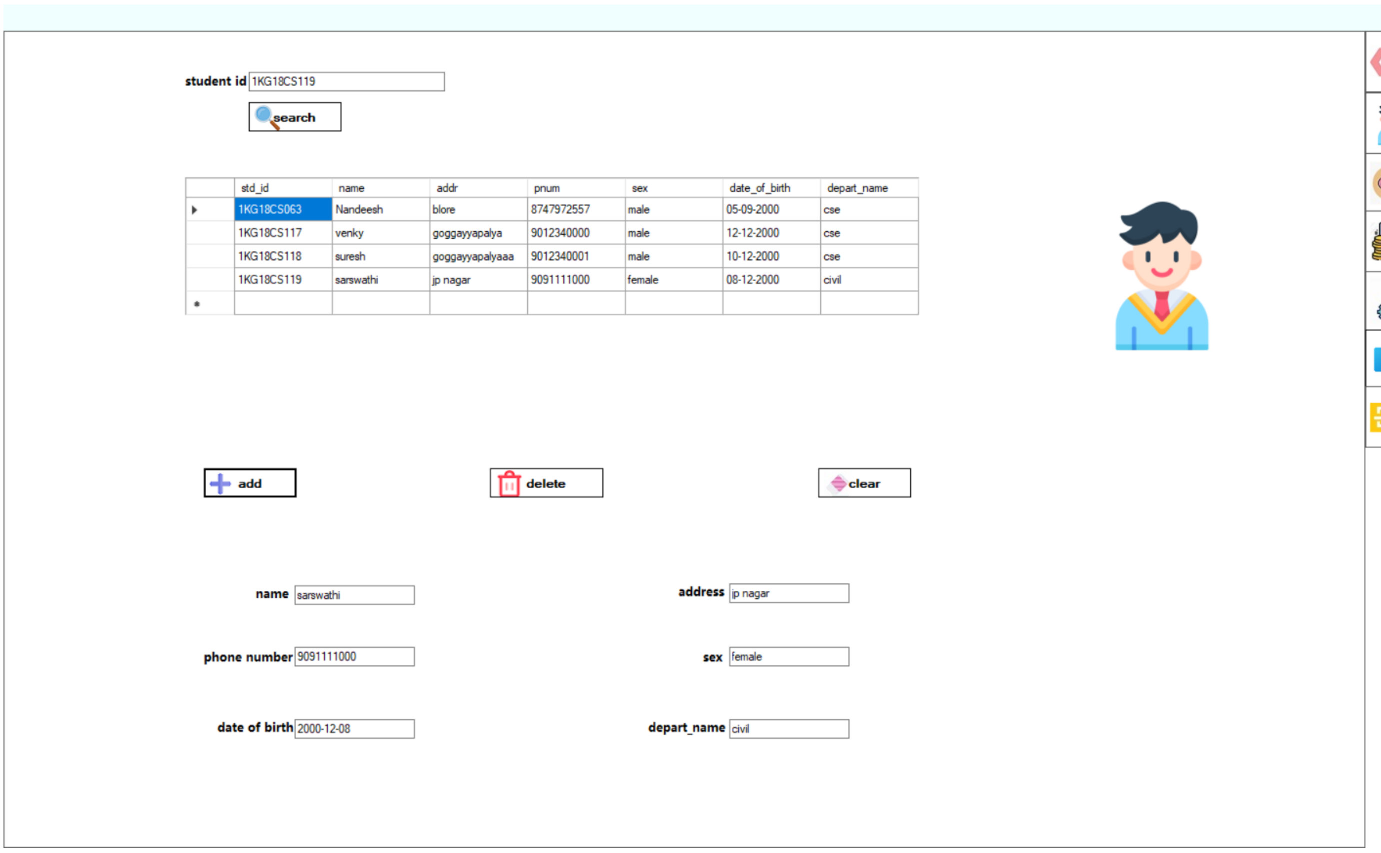


Fig 6.2: Homepage

6.3 STUDENT USER CONTROL



student id 1KG18CS119

	std_id	name	addr	pnum	sex	date_of_birth	depart_name
▸	1KG18CS063	Nandeesh	blore	8747972557	male	05-09-2000	cse
	1KG18CS117	venky	goggayyapalya	9012340000	male	12-12-2000	cse
	1KG18CS118	suresh	goggayyapalyaaa	9012340001	male	10-12-2000	cse
	1KG18CS119	sarswathi	jp nagar	9091111000	female	08-12-2000	civil
*							

name sarswathi

address jp nagar

phone number 9091111000

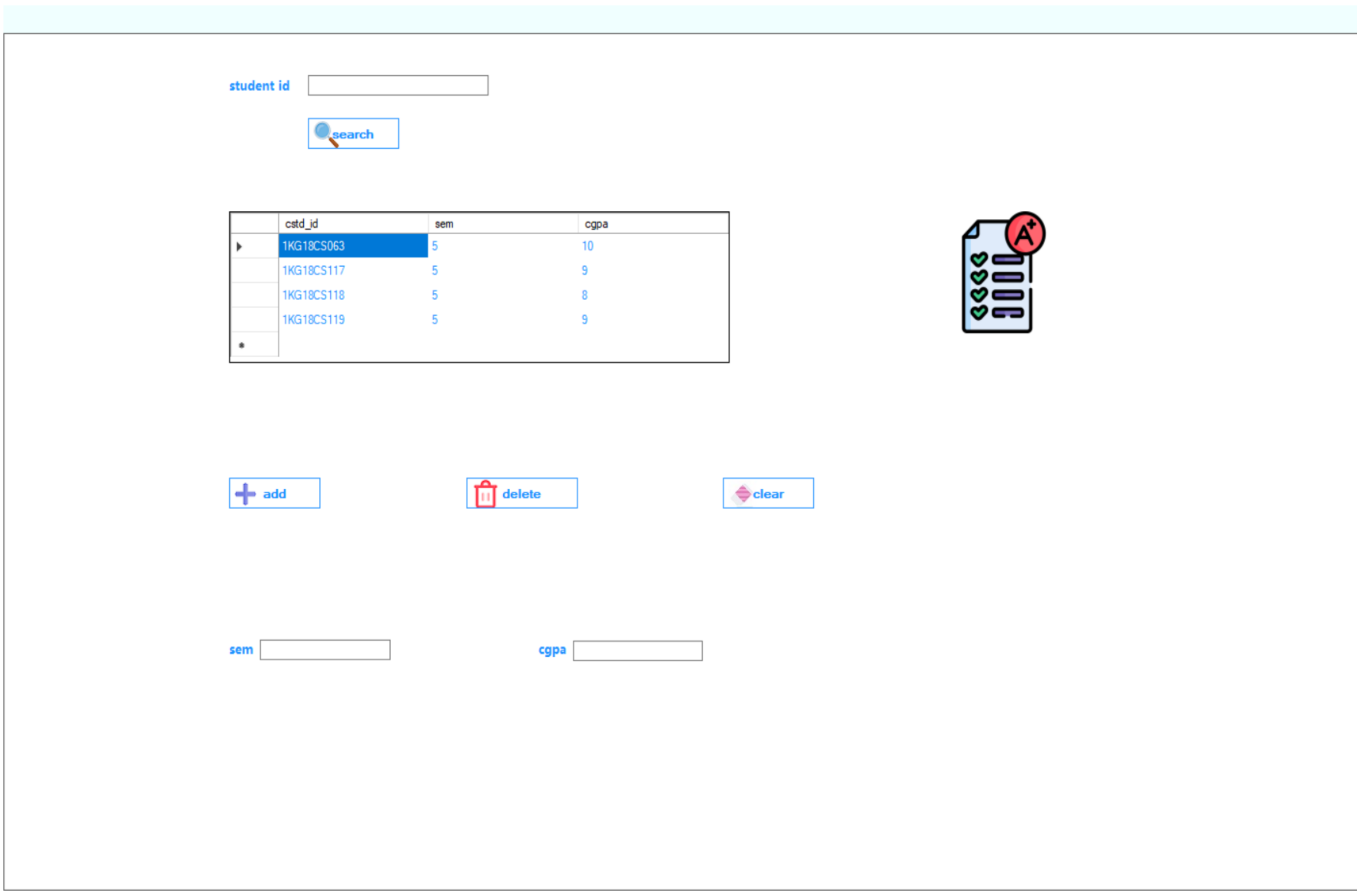
sex female

date of birth 2000-12-08

depart_name civil

Fig 6.3: Student user control

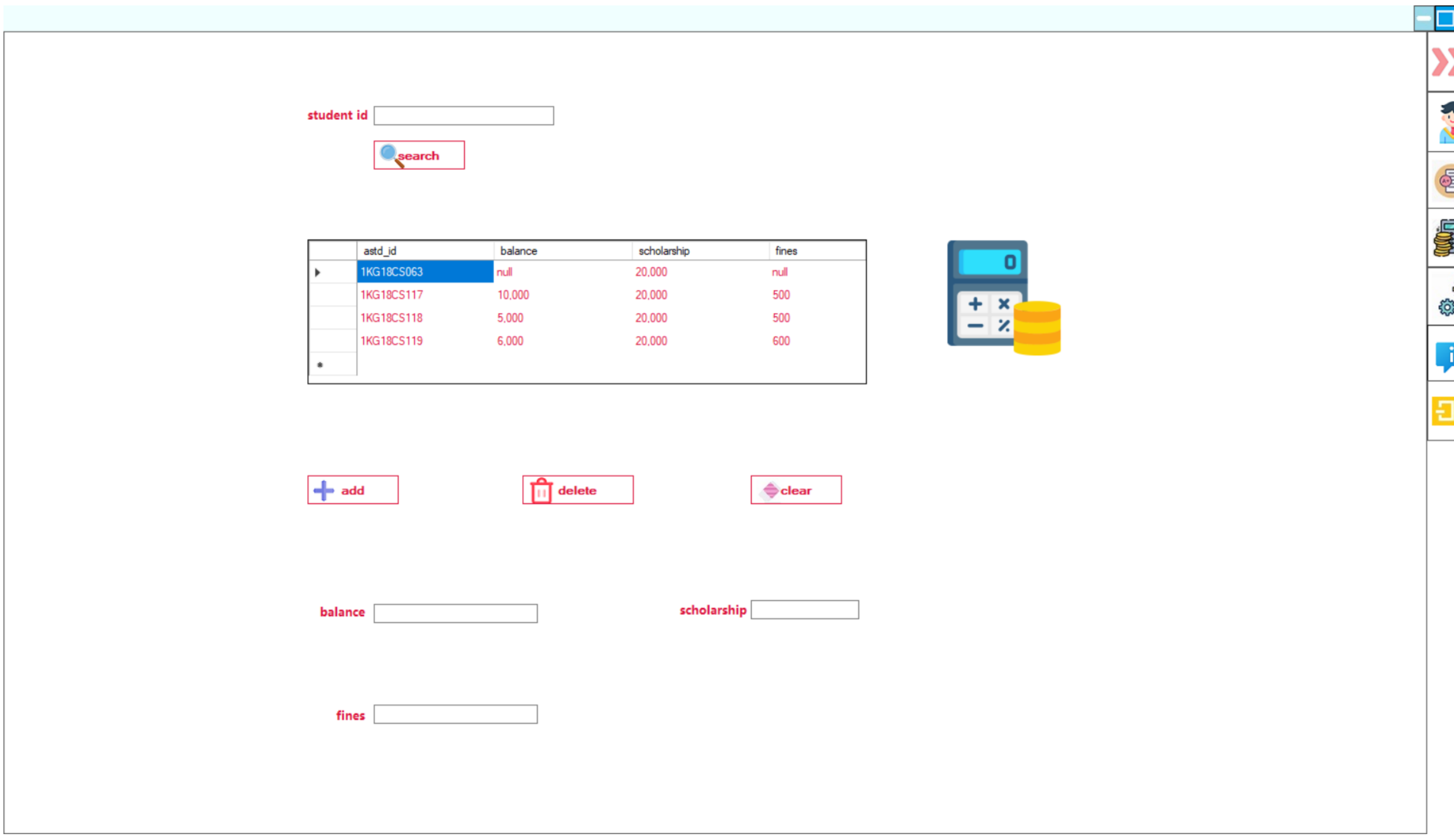
6.4 GRADE USER CONTROL



	cstd_id	sem	cgpa
▸	1KG18CS063	5	10
	1KG18CS117	5	9
	1KG18CS118	5	8
	1KG18CS119	5	9
*			

Fig 6.4: Grade user control

6.5 ACCOUNT USER CONTROL



astd_id	balance	scholarship	fines
1KG18CS063	null	20,000	null
1KG18CS117	10,000	20,000	500
1KG18CS118	5,000	20,000	500
1KG18CS119	6,000	20,000	600

Fig 6.5: Account user control

Chapter 7

CONCLUSION

The college management system provides easier and clean maintenance of the student data.

The admin can delete, add and update data. This project is to reduce the manual work of the administrator by helping in maintaining all the records computerised which is safe and secure.

This project does several things such as add, delete, update student data.

For the parents who queries about his child the retrieval of the particular student data is required among thousands of students. That is also successfully achieved through our project

- Centralized database
- Easier adding and delete of data
- User friendly environment.
- Efficient management of student data

REFERENCES

[1] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

[2] Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass: Addison-Wesley. Silberschatz A., Korth H. F., & Sudarshan S. (2011).

[3] Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.

[4] Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.