OBJECT ORIENTED PROGRAMMING WITH C#

21.1 PLYMOUTH

ACADEMIC YEAR 2022

SUTTLER – BADMINTON SCORE AND PLAYER MANAGEMENT SYSTEM REPORT

SEMESTER 2

GROUP NUMBER 17

Lecturer: Pramudya Thilakaratne

Acknowledgment

We would like to express our gratitude to everyone who contributes to this project's success. Also, we would want to convey our special thanks to everyone who helped us in completing this project on time.

We would like to express our gratitude to our module lecturer, Mr.Pramudya Thilakaratne, one of the lecturers at NSBM Green University's faculty of Computing, without whose support and guidance this project would not have been success.

Project Overview

Shuttler - Badminton Score and Player Management System employs file handling to store data such as player details and scores of each player. The program may show scores of each player and the admin can enter score updates of the players during a match.

The 'Shuttler' project is built on a database, C# object-oriented programming language, and network techniques. We used MySQL as the best and easiest way to store our information because we have many areas where we can keep records. The front-end software of Shuttler is C#, which is an object-oriented programming language that is linked to a MySQL server. Here to increase the scores of the players we used user controls in windows forms. Shuttler's necessary modules and features are custom-built to meet the requirements of each customer.

Index

1. System Introduction

The project ‘Shuttler’ is based on a badminton scoreboard in the university’s badminton matches.
The project Shuttler – Badminton Score and Player Management System is based on the players who play badminton and their scores. It includes registration of players which means player profile, storing their details into the system so an admin can enter the score of the registered player to the scoreboard during the match. Also, they can check the players using the student Id of each player.

2. Objectives

- To develop a system that allows admins to enter the player score during the match to the system which can easily sort out the players' marks anytime.
- To facilitate players to check their scores and to get a better understanding of their status.
  To track the participant's performance.
- To minimize the difficulties of finding scores of the players in previous matches.

3. System Analysis

3.1. As-is System

When considering the current sports level; sports clubs, schools, and universities follow a manual system for the scoreboard management system. NSBM also follows such a manual system in our university despite badminton being more popular there. With manual techniques and approaches, data contain human errors and can fluctuate. And no specific way to store data of the player and matches. So, our group planned to make an application to avoid these errors and make a database about the players and matches. Because most of our group members are badminton players, we faced them regularly. That’s why we came up with the system called Shuttler – The Badminton Score and Player Management System.

3.2. To-be System

Shuttler- Badminton Score and Player Management System is designed to replace the existing manual scoreboard system by using a digital platform. The new system is to store the player details and scores efficiently and reduce time, resources, and physical contact currently required for those needs. By using the Shuttler the sports council or badminton club administration can enter score updates of the players during a match and decide its suitability for the upcoming tournaments.

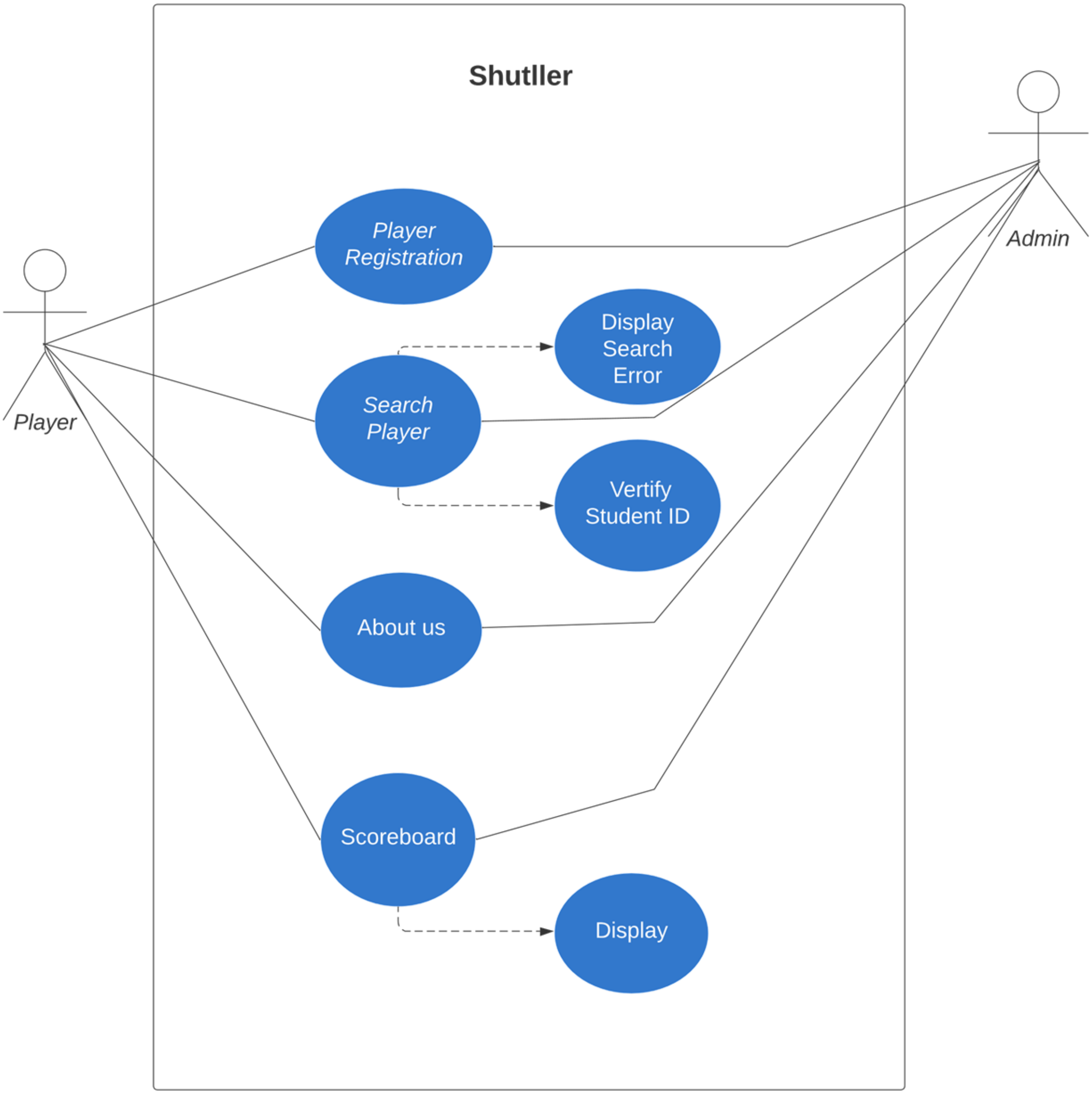
4. Software Requirements Specifications

Software requirements for the project:

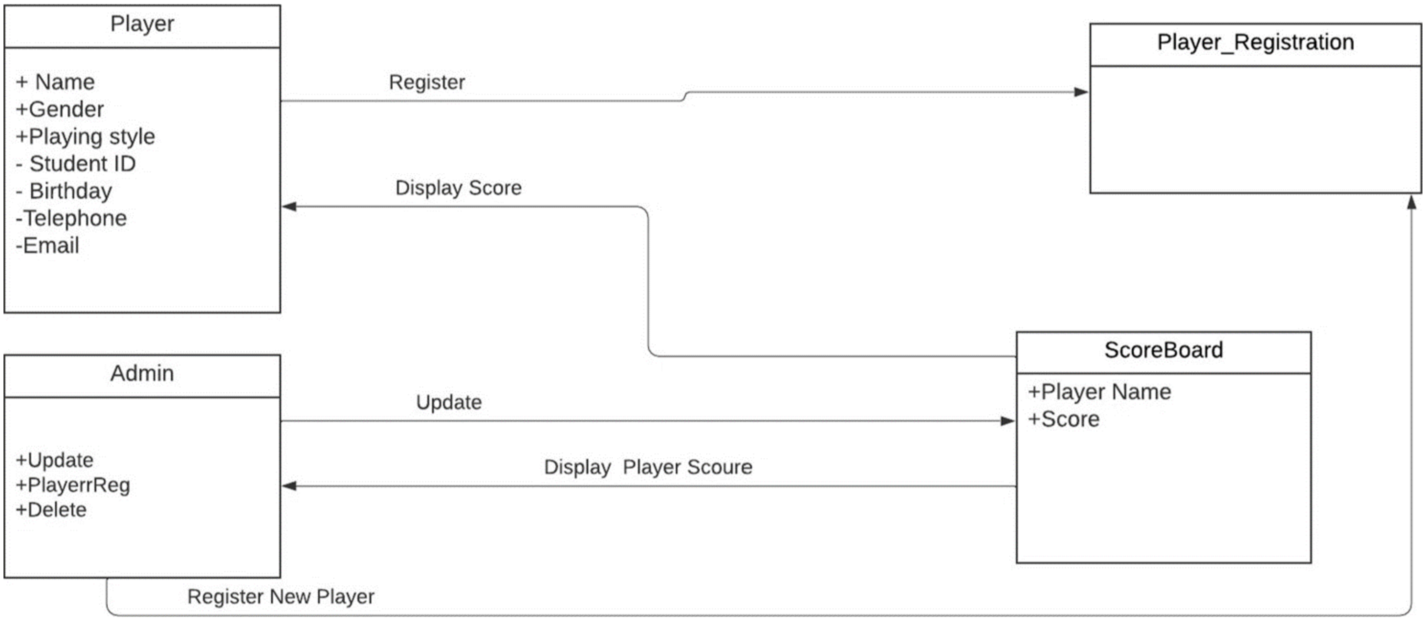
Operating System	: Windows 7/8/10
Main Software (IDE)	: Visual Studio 2022
Front End	: #C
Server-Side Script	: MySQL
Database	: MySQL
Framework	: Net Framework 4.7

5. System Design

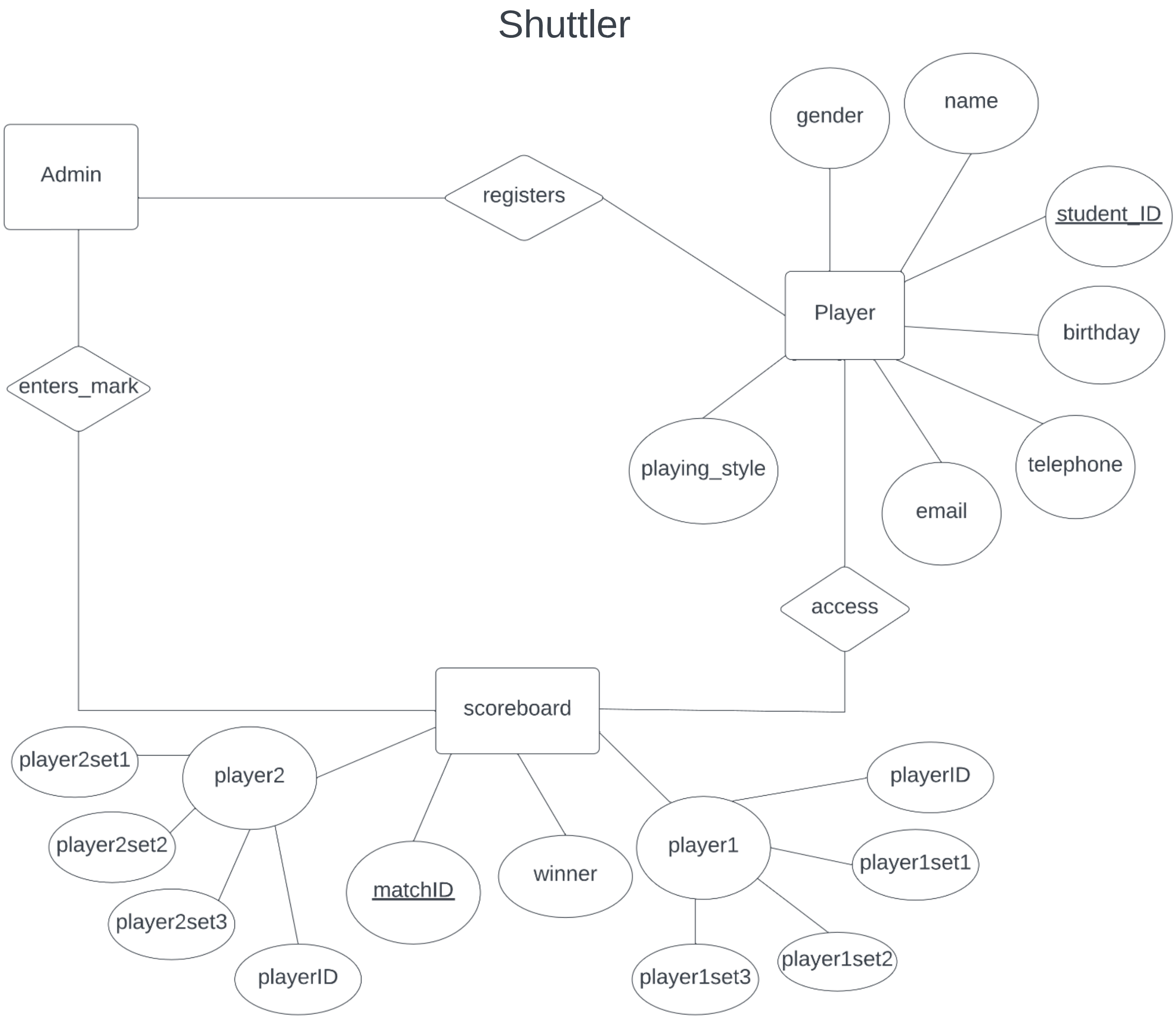
5.1. Use Case Diagram



5.2. Class Diagram



5.3. ER Diagram



Assumptions:

1. When players score, admin must increase the relevant score.
2. The score that belongs to the relevant set is called the player set.

6. System Implementation

6.1. Source Codes

AddPlayer.cs

```
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 last_final
{
    public partial class AddPlayer : Form
    {
        static string databaseName = "playerdb.mdf";

        string connectionString = @"Data Source(localdb)\MSSQLLocalDB;
AttachFilename=" + (Environment.CurrentDirectory) + @"\" + databaseName + ";
Integrated Security=True;";
        public AddPlayer()
        {

            InitializeComponent();
        }

        private void getAllPlayerData()
        {
        }

        public void save_Click(object sender, EventArgs e)
        {

            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            try
            {
                con.Open();
                SqlCommand cmd = con.CreateCommand();
                cmd.CommandType = CommandType.Text;
                //string sId = stdId.Text, sname = name.Text, sbirthday =
this.dateTimePicker1.Text, sgender = gender, stelephone = telephone.Text, semail =
email.Text, spstyle = pstyle;
                //cmd.CommandText = "insert into Player(stdId, name, birthday,
gender,telephone, email, pstyle) values('" + sId + "', '" + sname + "', '" +
sbirthday + "', '" + sgender + "', '" + stelephone + "', '" + semail + "', '" +
spstyle + "')";
```

```
                cmd.CommandText = "insert into Player(stdId, name, birthday, 
gender,telephone, email, pstyle) values('" + stdId.Text + "', '" + name.Text + "', 
'" + this.dateTimePicker1.Text + "', '" + gender + "', '" + telephone.Text + "', '" 
+ email.Text + "', '" + pstyle + "')";
                cmd.ExecuteNonQuery();
                //name.Text = "";

            }
            catch (SqlException)
            {
                Console.WriteLine("Error Generated! Please Check Again.");
            }
            //finally

            con.Close();
            MessageBox.Show("Data Successfully Added");
            this.Controls.Clear();
            this.InitializeComponent();

            //clearData objcl = new clearData();
            //objcl.clearForm();


        }

        string gender;
        private void male_CheckedChanged(object sender, EventArgs e)
        {
            gender = "Male";
        }

        private void female_CheckedChanged(object sender, EventArgs e)
        {
            gender = "Female";
        }


        string pstyle;
        private void lhand_CheckedChanged(object sender, EventArgs e)
        {
            pstyle = "Left Hander";
        }

        private void rhand_CheckedChanged(object sender, EventArgs e)
        {
            pstyle = "Right Hander";
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
       }
```

Form1.cs

```
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;

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

        private void about_Click(object sender, EventArgs e)
        {
            //userControl11.Hide();
            //userControl21.Show();
            AboutBox1 aboutBox1 = new AboutBox1();
            aboutBox1.StartPosition = FormStartPosition.CenterScreen;
            aboutBox1.Show();
        }

        private void shuttler_Click(object sender, EventArgs e)
        {
            userControl11.Show();
            userControl21.Hide();
        }

        private void userControl11_Load(object sender, EventArgs e)
        {

        }
    }
       }
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace last_final
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

ReadForm.cs

```
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 last_final
{
    public partial class ReadForm : Form
    {
        public ReadForm()
        {
            InitializeComponent();
        }

        private void search_Click(object sender, EventArgs e)
        {

            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            SqlDataAdapter adap;

            try
            {
                con.Open();

                string userIn = searchBox.Text;


                rstdId.Text = userIn;


                //name
                string cmdst1 = "SELECT name FROM Player WHERE stdId = '" + userIn +
"'";

                adap = new SqlDataAdapter(cmdst1, con);
                DataSet ds1 = new DataSet();
                adap.Fill(ds1);
                rname.Text = ds1.Tables[0].Rows[0][0].ToString();

                //age
                string cmdst2 = "SELECT FLOOR(DATEDIFF(DAY, (SELECT birthday FROM
Player WHERE stdId = '" + userIn + "'), GETDATE()) / 365.25) FROM Player WHERE stdId
= '" + userIn + "'";
                adap = new SqlDataAdapter(cmdst2, con);
                DataSet ds2 = new DataSet();
                adap.Fill(ds2);
                rage.Text = ds2.Tables[0].Rows[0][0].ToString();
                //telephone
                string cmdst3 = "SELECT telephone FROM Player WHERE stdId = '" +
userIn + "'";
                adap = new SqlDataAdapter(cmdst3, con);
                DataSet ds3 = new DataSet();
                adap.Fill(ds3);
                rtelephone.Text = ds3.Tables[0].Rows[0][0].ToString();

                //playing style
                string cmdst4 = "SELECT pstyle FROM Player WHERE stdId = '" + userIn
+ "'";
                adap = new SqlDataAdapter(cmdst4, con);
                DataSet ds4 = new DataSet();
                adap.Fill(ds4);
                rpstyle.Text = ds4.Tables[0].Rows[0][0].ToString();


                //winning perc
                string cmdst52 = "SELECT COUNT(*) FROM match WHERE player1 = '" +
userIn + "' OR player2 = '" + userIn + "'";
                string cmdst53 = "SELECT COUNT(*) FROM match WHERE win = '" + userIn
+ "'";

                adap = new SqlDataAdapter(cmdst52, con);
                DataSet ds52 = new DataSet();
                adap.Fill(ds52);

                adap = new SqlDataAdapter(cmdst53, con);
                DataSet ds53 = new DataSet();
                adap.Fill(ds53);


                string totalm = ds52.Tables[0].Rows[0][0].ToString();
                string winm = ds53.Tables[0].Rows[0][0].ToString();

                //double avg = (Double.Parse(winm) / Double.Parse(totalm)) * 100;
                double avg = Math.Round((Double.Parse(winm) / Double.Parse(totalm))
* 100,2);

                rwpercentage.Text = $"{avg}%";

                //Console.WriteLine(avg);



                //current rank
                //string cmdst6 = "SELECT crank FROM Player WHERE stdId = '" +
userIn + "'";
                //adap = new SqlDataAdapter(cmdst6, con);
                //DataSet ds6 = new DataSet();
                //adap.Fill(ds6);
                //rcrank.Text = ds6.Tables[0].Rows[0][0].ToString();


                //gender
```

```
        string cmdst7 = "SELECT gender FROM Player WHERE stdId = '" + userIn
+ "'";
        adap = new SqlDataAdapter(cmdst7, con);
        DataSet ds7 = new DataSet();
        adap.Fill(ds7);
        rgender.Text = ds7.Tables[0].Rows[0][0].ToString();


        //email
        string cmdst8 = "SELECT email FROM Player WHERE stdId = '" + userIn
+ "'";
        adap = new SqlDataAdapter(cmdst8, con);
        DataSet ds8 = new DataSet();
        adap.Fill(ds8);
        remail.Text = ds8.Tables[0].Rows[0][0].ToString();


        //total matches
        string cmdst9 = "SELECT COUNT(*) FROM match WHERE player1 = '" +
userIn + "' OR player2 = '" + userIn + "'";
        adap = new SqlDataAdapter(cmdst9, con);
        DataSet ds9 = new DataSet();
        adap.Fill(ds9);
        rtmatches.Text = ds9.Tables[0].Rows[0][0].ToString();


        //highest rank
        //string cmdst10 = "SELECT hrank FROM Player WHERE stdId = '" +
userIn + "'";
        //adap = new SqlDataAdapter(cmdst10, con);
        //DataSet ds10 = new DataSet();
        //adap.Fill(ds10);
        //rhrank.Text = ds10.Tables[0].Rows[0][0].ToString();

        pictureBox1.Hide();
        panel1.Hide();
        //updateUC updateUC = new updateUC();
        //updateUC.Hide();

      }

      catch (IndexOutOfRangeException)
      {
        MessageBox.Show("Please Enter a valid Student Id");
      }

      catch (SqlException)
      {
       //MessageBox.Show($"{ex}");
       MessageBox.Show("Error !");
      }



      con.Close();

    }
```

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

        }

        int count = 0;
        private void timer1_Tick(object sender, EventArgs e)
        {
            if (count < 5) //change the number of images
            {
                pictureBox1.Image = imageList1.Images[count];
                count++;
            }
            else
            {
                count = 0;
            }
        }

        private void delete_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");

            try
            {
                con.Open();
                string userIn = searchBox.Text;
                string sql = "DELETE FROM Player WHERE stdId = '" + userIn + "'";
                SqlCommand cmd = new SqlCommand(sql, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Player Profile Deleted Successfully");
            }

            catch
            {
                MessageBox.Show("Error !");
            }
            con.Close();
            panel1.Show();
            pictureBox1.Show();

        }

        private void edit_Click(object sender, EventArgs e)
        {

            panel1.Show();
            pictureBox1.Hide();
            updateForm updateUC = new updateForm();
            updateUC.Show();
            updateUC.BringToFront();



        }
        private void edit_Click_1(object sender, EventArgs e)
        {


            pictureBox1.Hide();
            updateForm updateUC = new updateForm();
            updateUC.Show();
            updateUC.BringToFront();
            updateUC.Visible = true;
        }

        private void delete_Click_1(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");

            try
            {
                con.Open();
                string userIn = searchBox.Text;
                string sql = "DELETE FROM Player WHERE stdId = '" + userIn + "'";
                SqlCommand cmd = new SqlCommand(sql, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Player Profile Deleted Successfully");
            }

            catch
            {
                MessageBox.Show("Error !");
            }
            con.Close();
            panel1.Show();
            pictureBox1.Show();
        }

        private void edit_Click_2(object sender, EventArgs e)
        {
            updateForm updateUC = new updateForm();
            updateUC.StartPosition = FormStartPosition.CenterScreen;
            updateUC.ShowDialog();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void timer1_Tick_1(object sender, EventArgs e)
        {

        }
    }
}
```

score.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace last_final
{
    public abstract class score
    {
        public abstract void scr(Label x);
    }
}
```

Scoreboard.cs

```
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 last_final
{
    public partial class Scoreboard : Form
    {
        string winner;
        bool thirdSet;
        public Scoreboard()
        {
            InitializeComponent();
        }

        private void btnScrL_Click(object sender, EventArgs e)
        {

            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");

            try
            {
                con.Open();
                SqlDataAdapter adap;
                string cmdst = "select name from Player where stdId ='" + 
txtPnL.Text + "'";
                adap = new SqlDataAdapter(cmdst, con);
                DataSet ds = new DataSet();
                adap.Fill(ds);
                lblPnameL.Text = ds.Tables[0].Rows[0][0].ToString();
            }
            catch (IndexOutOfRangeException)
            {
                MessageBox.Show("Please Enter a valid Student Id");
            }

            con.Close();

            lblPnameL.Visible = true;
        }

        private void btnScrR_Click(object sender, EventArgs e)
        {

            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
```

```
            try
            {
                SqlDataAdapter adap;
                con.Open();

                string cmdst = "select name from Player where stdId ='" +
txtPnR.Text + "'";
                adap = new SqlDataAdapter(cmdst, con);
                DataSet ds = new DataSet();
                adap.Fill(ds);
                lblPnameR.Text = ds.Tables[0].Rows[0][0].ToString();
            }
            catch (IndexOutOfRangeException)
            {
                MessageBox.Show("Please Enter a valid Student Id");
            }

            con.Close();

            lblPnameR.Visible = true;
        }

        private void lblDecL_Click(object sender, EventArgs e)
        {
            score scr = new ScoreDec();
            scr.scr(lblSL);
            //int iCount = Convert.ToInt32(lblSL.Text);
            //if (iCount > 0)
            //{
            //    for (iCount = Convert.ToInt32(lblSL.Text); iCount >= 0;)
            //    {

            //        iCount--;
            //        lblSL.Text = iCount.ToString();
            //        break;

            //    }
            //}
        }

        private void btnInl_Click(object sender, EventArgs e)
        {
            score scr = new ScoreInc();
            scr.scr(lblSL);

            //int iCount = Convert.ToInt32(lblSL.Text);

        }

        private void btnClrL_Click(object sender, EventArgs e)
        {
            //lblWon.Visible = false;
            lblSL.Text = "0";
            //lblSetL.Text = "0";
        }

        private void lblDecR_Click(object sender, EventArgs e)
        {
            score scr = new ScoreDec();
            scr.scr(lblSR);
            //int iCount = Convert.ToInt32(lblSR.Text);

        }

        private void btnIn_Click(object sender, EventArgs e)
        {
            score scr = new ScoreInc();
            scr.scr(lblSR);

        }

        private void btnClR_Click(object sender, EventArgs e)
        {
            //lblWon.Visible = false;
            lblSR.Text = "0";
            //lblSetR.Text = "0";
        }

        private void btnNew_Click(object sender, EventArgs e)
        {
            this.Controls.Clear();
            this.InitializeComponent();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");

            con.Open();
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "insert into match values('" + setName.Text + "', '" +
s1lbl3.Text + "', '" + s2lbl3.Text + "', '" + s3lbl3.Text + "', '" + s1lbl4.Text +
"', '" + s2lbl4.Text + "', '" + s3lbl4.Text + "', '" + txtPnL.Text + "', '" +
txtPnR.Text + "', '" + lblIndex.Text + "')";
            //cmd.CommandText = "insert into match(p1s1, p1s2, p1s3, p2s1, p2s2,
p2s3, player1, player2, win) values('" + s1lbl3.Text + "', '" + s2lbl3.Text + "', '"
+ s3lbl3.Text + "', '" + s1lbl4.Text + "', '" + s2lbl4.Text + "', '" + s3lbl4.Text +
"', '" + txtPnL.Text + "', '" + txtPnR.Text + "', '" + lblIndex.Text + "')";
            cmd.ExecuteNonQuery();

            //-----------------------------------------

            //-----------------------------------------

            con.Close();

            MessageBox.Show("Data Successfully Added");
        }

        private void button8_Click(object sender, EventArgs e)
        {
            int sl = Convert.ToInt32(lblSL.Text);
            int sr = Convert.ToInt32(lblSR.Text);
            int setL = Convert.ToInt32(lblSetL.Text);
            int setR = Convert.ToInt32(lblSetR.Text);
            int count = 0;

            if ((sl >= 21) || (sr >= 21))
            {

                if (sr < sl)
                {
                    for (setL = Convert.ToInt32(lblSetL.Text); setL < 2;)
                    {

                        setL++;
                        lblSetL.Text = setL.ToString();
                        break;

                    }
                }


                if (sl < sr)
                {
                    for (setR = Convert.ToInt32(lblSetR.Text); setR < 2;)
                    {

                        setR++;
                        lblSetR.Text = setR.ToString();
                        break;

                    }
                }

                if (set1.Visible == true)
                {

                    count = 2;

                }
                else
                {
                    if ((setL == 1) || (setR == 1))
                    {
                        count = 1;
                        if (count == 1)
                        {
                            set1.Visible = true;
                            // set1.Text = setName.Text;
                            s1lbl1.Text = lblPnameL.Text;
                            s1lbl2.Text = lblPnameR.Text;
                            s1lbl3.Text = lblSL.Text;
                            s1lbl4.Text = lblSR.Text;
                        }
                    }
                }

                if (set2.Visible == true)
                {
                    if (thirdSet == true)
                    {
                        set3.Visible = true;
                        // set3.Text = setName.Text;
                        s3lbl1.Text = lblPnameL.Text;
                        s3lbl2.Text = lblPnameR.Text;
                        s3lbl3.Text = lblSL.Text;
                        s3lbl4.Text = lblSR.Text;
                    }
                    //else
                    //{

                    //}

                    //count = 3;
                    //if (count == 3)
                    //{
                    //    set3.Visible = true;
                    //   // set3.Text = setName.Text;
                    //    s3lbl1.Text = txtPnL.Text;
                    //    s3lbl2.Text = txtPnR.Text;
                    //    s3lbl3.Text = lblSL.Text;
                    //    s3lbl4.Text = lblSR.Text;
                    //}

                }
                else
                {
                    if ((setL == 2) || ((setL == 1) && (setR == 1)))
                    {
                        count = 2;
                        //lblWon.Visible = true;
                        //lblWon.Text = "WINNER : " + txtPnL.Text;
                        if (count == 2)
                        {
                            set2.Visible = true;
                            //set2.Text = setName.Text;
                            s2lbl1.Text = lblPnameL.Text;
                            s2lbl2.Text = lblPnameR.Text;
                            s2lbl3.Text = lblSL.Text;
                            s2lbl4.Text = lblSR.Text;
                        }
                    }
                    if ((setR == 2) || ((setL == 1) && (setR == 1)))
                    {
                        count = 2;
                        //lblWon.Visible = true;
                        //lblWon.Text = "WINNER : " + txtPnR.Text;
                        if (count == 2)
                        {
                            set2.Visible = true;
                            //set2.Text = setName.Text;
                            s2lbl1.Text = lblPnameL.Text;
                            s2lbl2.Text = lblPnameR.Text;
                            s2lbl3.Text = lblSL.Text;
                            s2lbl4.Text = lblSR.Text;
                        }
                    }
                    if ((setL == 1) && (setR == 1)) { thirdSet = true; }
                }


                if (setL == 2)
                {
                    winner = lblPnameL.Text;
                    lblWon.Visible = true;
                    lblWon.Text = "WINNER : " + winner;
                    lblIndex.Text = txtPnL.Text;
                }
                if (setR == 2)
                {
                    winner = lblPnameR.Text;
                    lblWon.Visible = true;
                    lblWon.Text = "WINNER : " + winner;
                    lblIndex.Text = txtPnR.Text;
                }
                //-----------------------------------

            }
        }
    }
}
```

ScoreL.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace last_final
{
    public class ScoreInc : score
    {
        override public void scr(Label x)
        {
            int iCount = Convert.ToInt32(x.Text);
            if (iCount < 30)
            {
                for (iCount = Convert.ToInt32(x.Text); iCount <= 30;)
                {

                    iCount++;
                    x.Text = iCount.ToString();
                    break;

                }
            }
        }

    }
    public class ScoreDec : score
    {
        override public void scr(Label x)
        {
            int iCount = Convert.ToInt32(x.Text);
            if (iCount > 0)
            {
                for (iCount = Convert.ToInt32(x.Text); iCount >= 0;)
                {

                    iCount--;
                    x.Text = iCount.ToString();
                    break;

                }
            }
        }
    }
}
```

updateForm.cs

```
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 last_final
{
    public partial class updateForm : Form
    {
        public updateForm()
        {
            InitializeComponent();
        }

        private void update_Click(object sender, EventArgs e)
        {

            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            con.Open();


            string userIn = textBox1.Text;

            string newname = name.Text;



            string Query = "UPDATE Player SET name = '" + newname + "' WHERE stdId =
'" + userIn + "' ";

            SqlCommand cmd = new SqlCommand(Query, con);
            cmd.ExecuteNonQuery();

            con.Close();
            MessageBox.Show("Player Name Updated Successfully");


        }

        private void search_Click(object sender, EventArgs e)
        {

        }



        private void update3_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            con.Open();


            string userIn = textBox1.Text;

            string newemail = email.Text;

            string Query = "UPDATE Player SET email = '" + newemail + "' WHERE stdId 
= '" + userIn + "' ";

            SqlCommand cmd = new SqlCommand(Query, con);
            cmd.ExecuteNonQuery();

            con.Close();
            MessageBox.Show("Player Email Updated Successfully");
        }



        private void update2_Click_1(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            con.Open();


            string userIn = textBox1.Text;

            string newtelephone = telephone.Text;

            string Query = "UPDATE Player SET telephone = '" + newtelephone + "' 
WHERE stdId = '" + userIn + "' ";

            SqlCommand cmd = new SqlCommand(Query, con);
            cmd.ExecuteNonQuery();

            con.Close();
            MessageBox.Show("Player Contact Number Updated Successfully");
        }

        private void update3_Click_1(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\ashan\Documents\playerdb.mdf
;Integrated Security=True;Connect Timeout=30");
            con.Open();


            string userIn = textBox1.Text;

            string newemail = email.Text;

            string Query = "UPDATE Player SET email = '" + newemail + "' WHERE stdId 
= '" + userIn + "' ";
```

```
            SqlCommand cmd = new SqlCommand(Query, con);
            cmd.ExecuteNonQuery();

            con.Close();
            MessageBox.Show("Player Email Updated Successfully");
        }
    }
}
```

userControl1.cs

```
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;

namespace last_final
{
    public partial class UserControl1 : UserControl
    {
        public UserControl1()
        {
            InitializeComponent();
        }

        int count = 0;
        private void timer1_Tick(object sender, EventArgs e)
        {
            if (count < 3) //change the number of images
            {
                pictureBox1.Image = imageList1.Images[count];
                count++;
            }
            else
            {
                count = 0;
            }
        }

        private void search_Click(object sender, EventArgs e)
        {
            ReadForm readForm = new ReadForm();
            readForm.StartPosition = FormStartPosition.CenterScreen;
            readForm.ShowDialog();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            if (count < 3) //change the number of images
            {
                pictureBox1.Image = imageList1.Images[count];
                count++;
            }
            else
            {
                count = 0;
            }
        }

        private void player_Click(object sender, EventArgs e)
        {
            AddPlayer addPlayer = new AddPlayer();
            addPlayer.StartPosition = FormStartPosition.CenterScreen;
            addPlayer.ShowDialog();
        }

        private void scoreboard_Click(object sender, EventArgs e)
        {
            Scoreboard scoreboard = new Scoreboard();
            scoreboard.StartPosition = FormStartPosition.CenterScreen;
            scoreboard.ShowDialog();

        }
    }
}
```

userContol2.cs

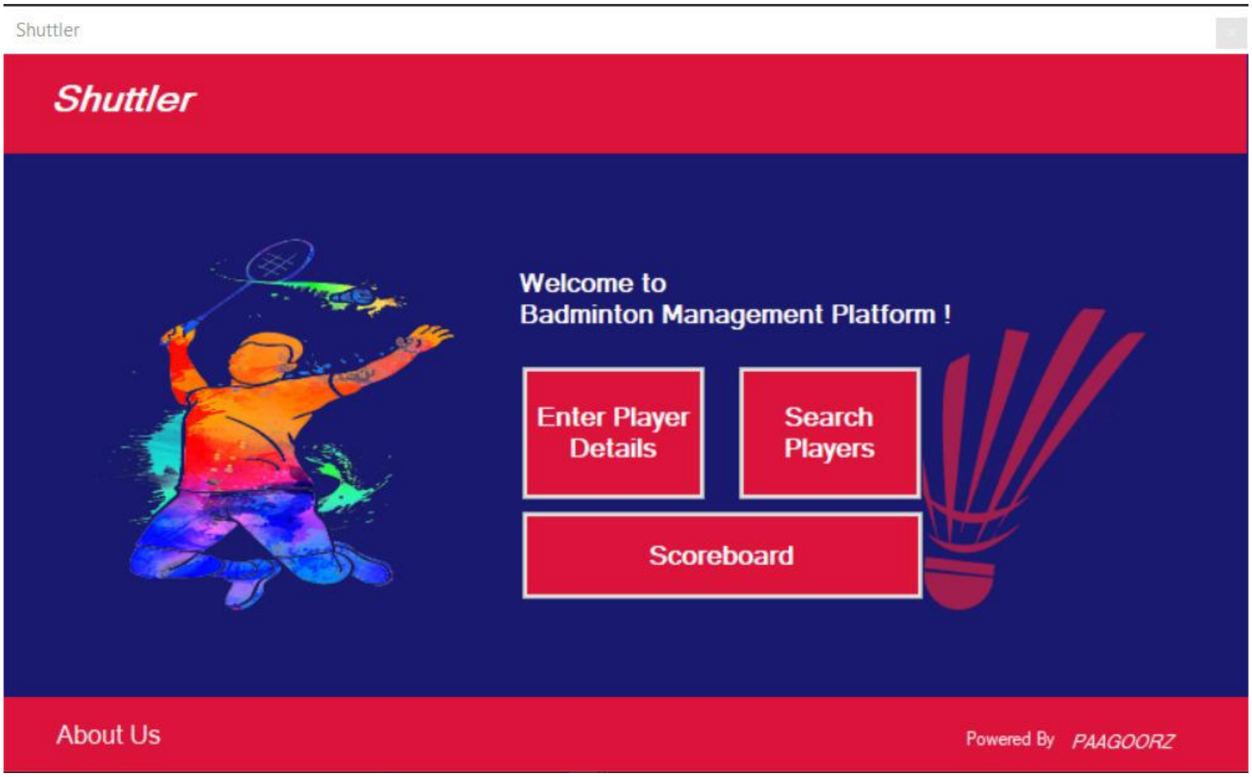
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace last_final
{
    public partial class UserControl2 : UserControl
    {
        public UserControl2()
        {
            InitializeComponent();
        }
    }
}
```

userControl3.cs

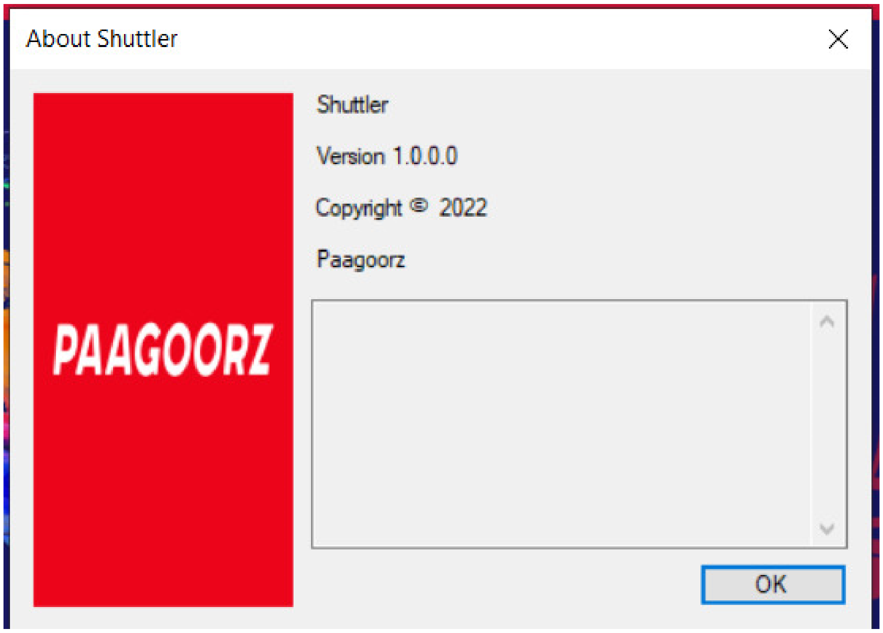
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace last_final
{
    public partial class UserControl3 : last_final.UserControl2
    {
        public UserControl3()
        {
            InitializeComponent();
        }
    }
}
```

7. Sample Screenshots

Home Window



About option



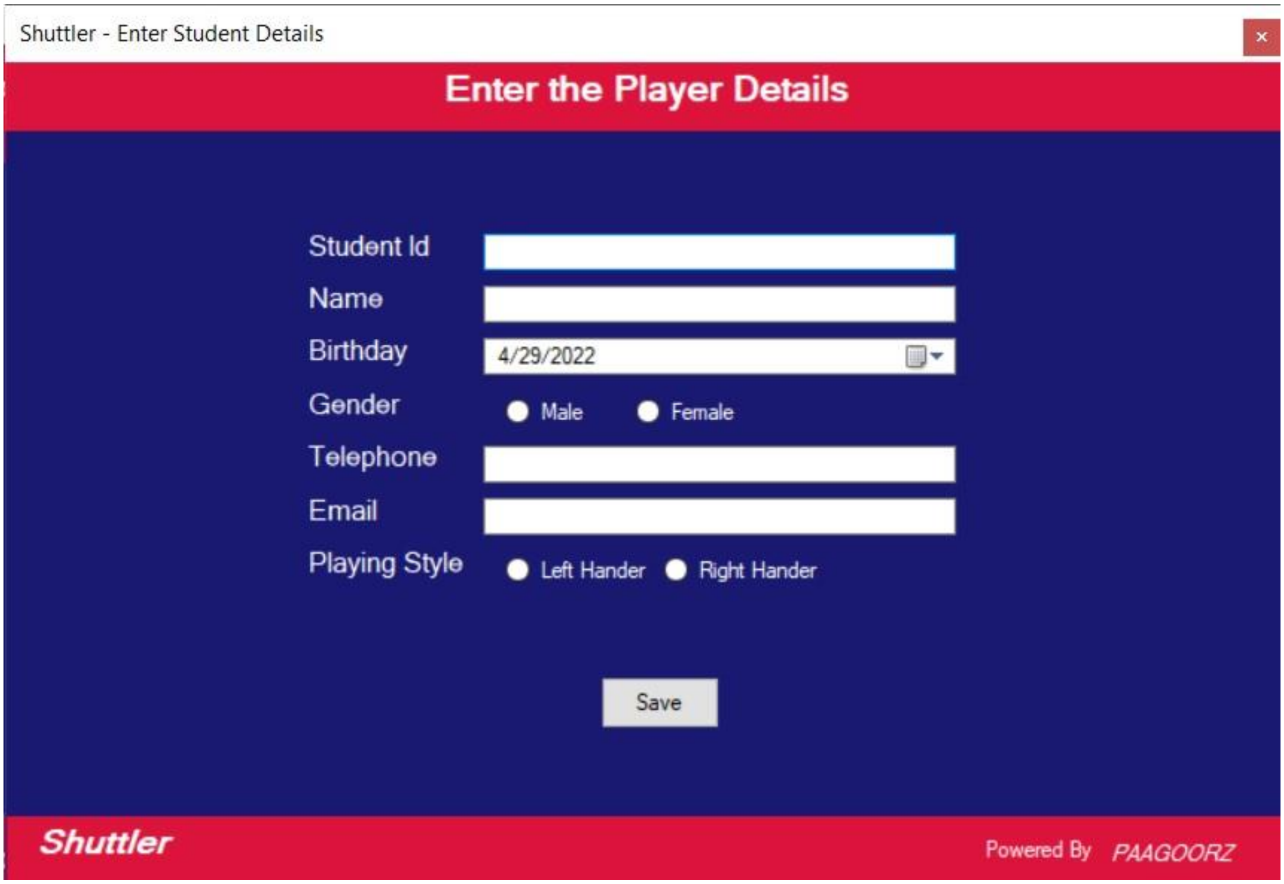
Shuttler

Version 1.0.0.0

Copyright © 2022

Paagoorz

Enter Player Details option



Shuttler - Enter Student Details

Enter the Player Details

Student Id

Name

Birthday 4/29/2022

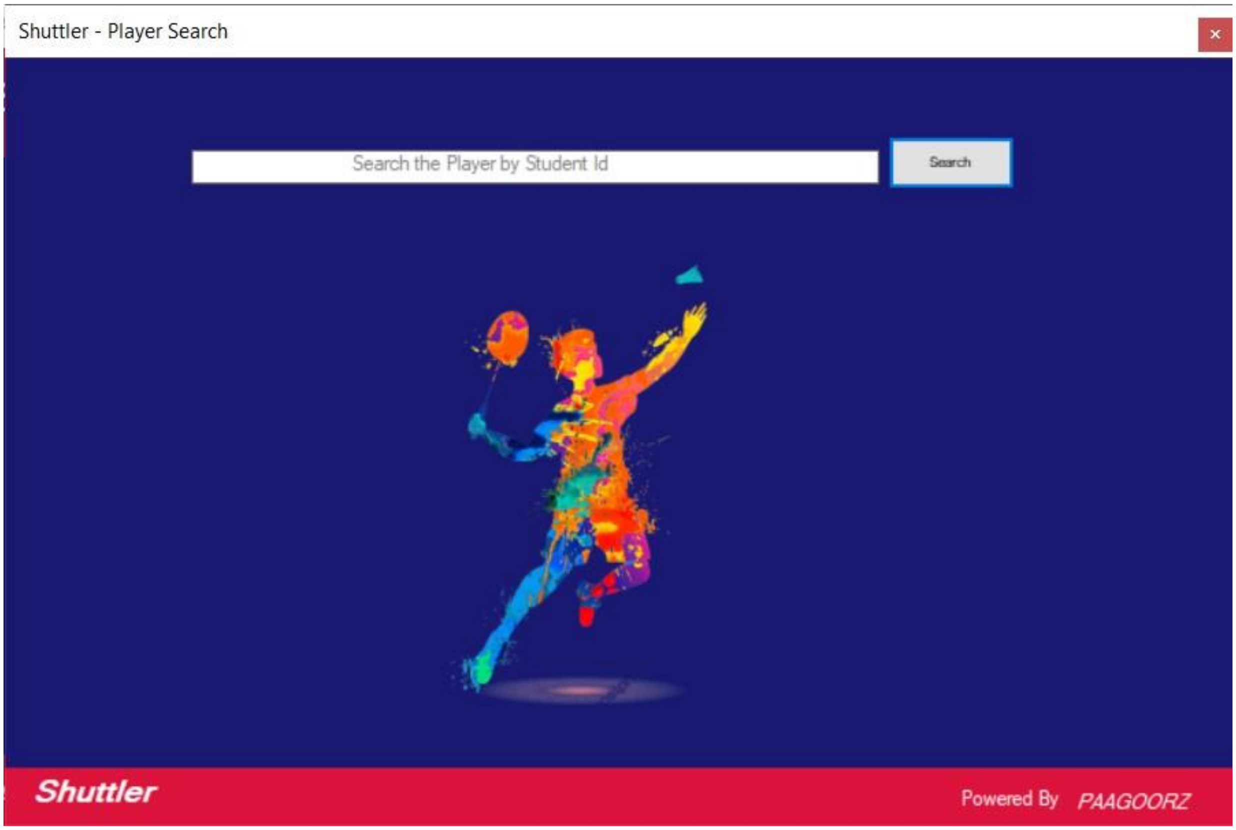
Gender ○ Male ○ Female

Telephone

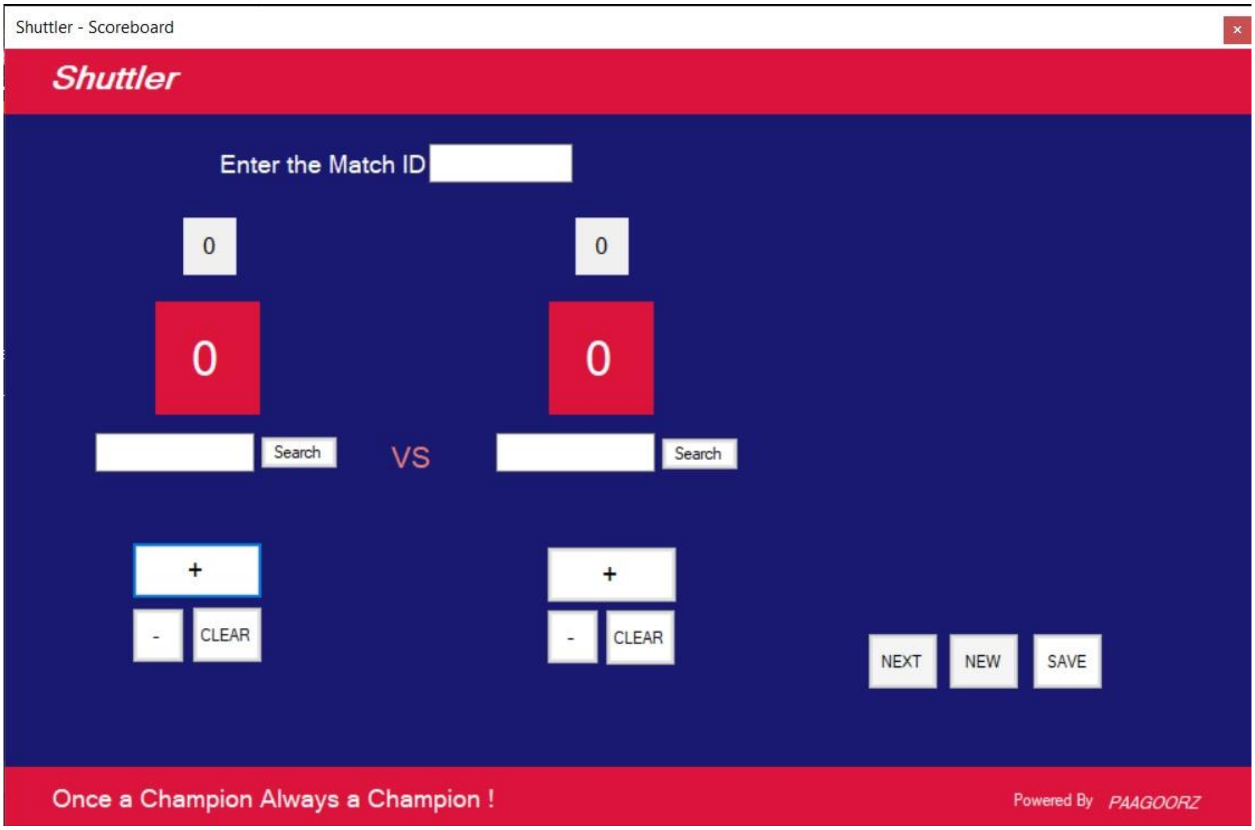
Email

Playing Style ○ Left Hander ○ Right Hander

Search Player option

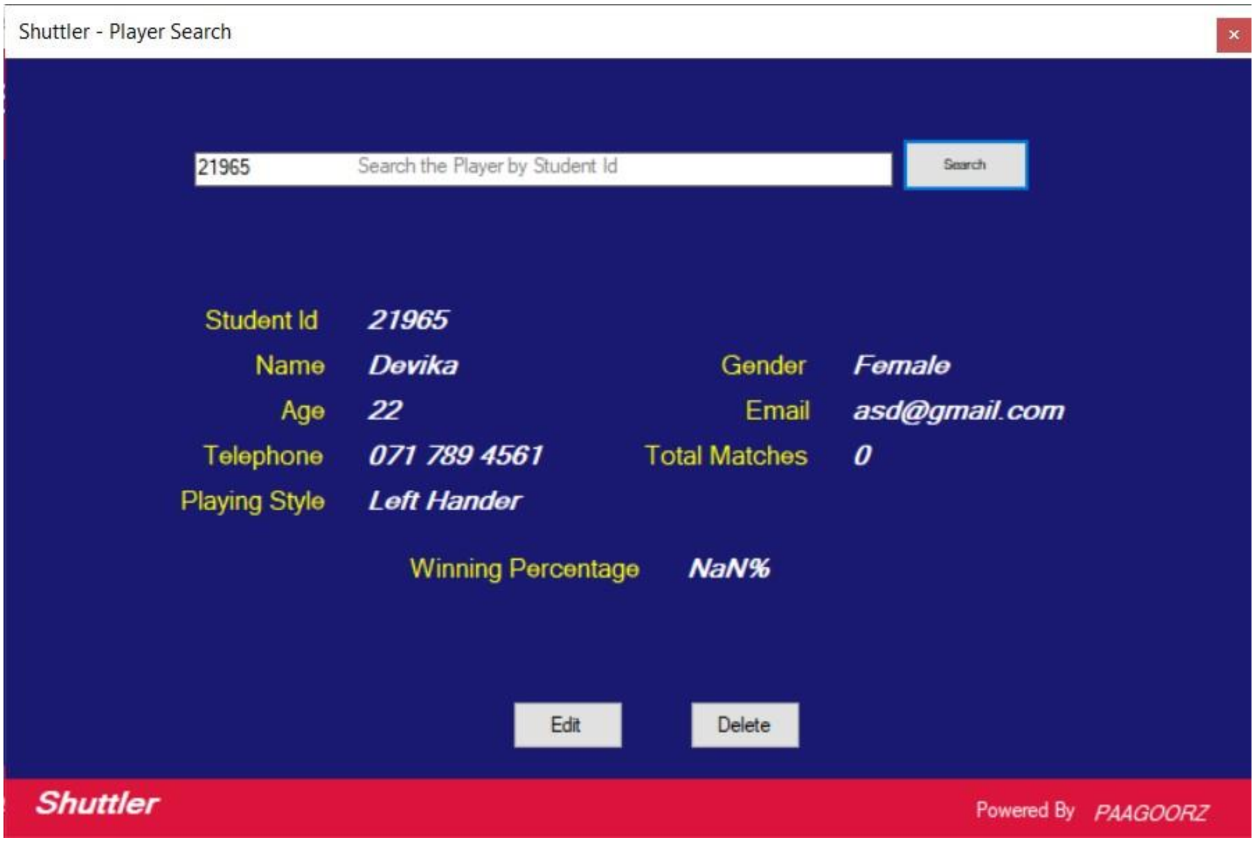


Scoreboard



Shuttler - Scoreboard
×
Shuttler
Enter the Match ID
0
0
0
0
Search
VS
Search
+
+
-
CLEAR
-
CLEAR
NEXT
NEW
SAVE
Once a Champion Always a Champion !
Powered By PAAGOORZ

Sample search result



21965

Student Id 21965

Name Devika

Gender Female

Age 22

Email asd@gmail.com

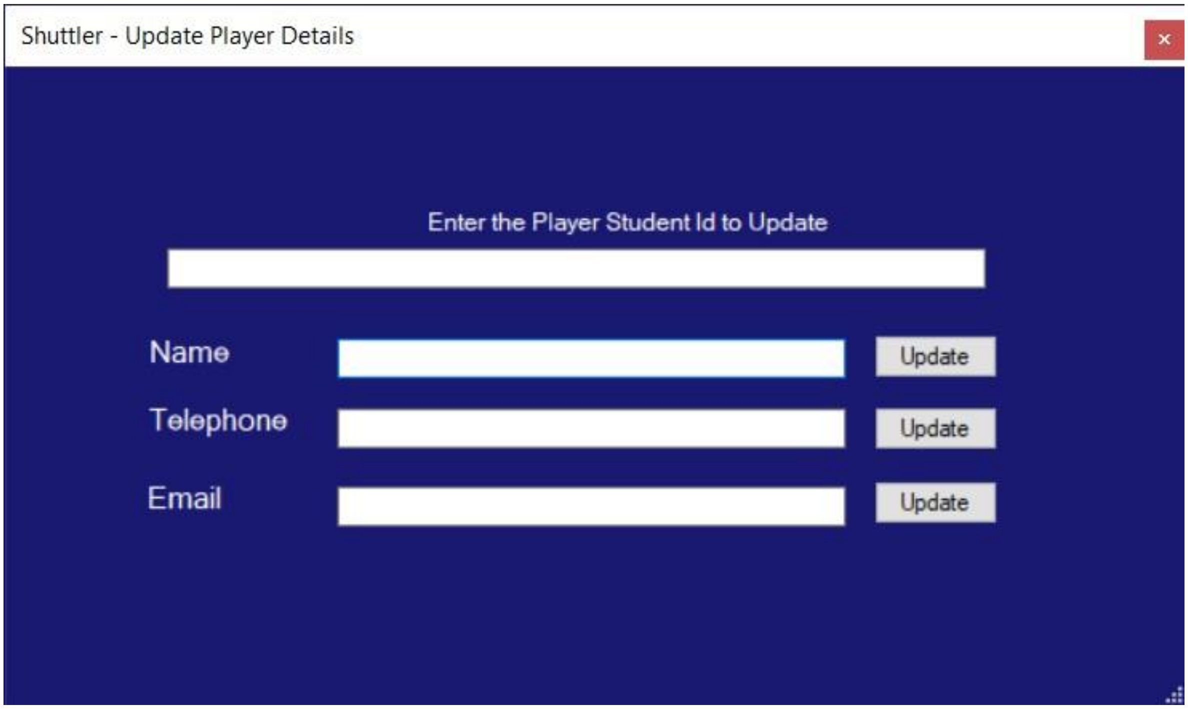
Telephone 071 789 4561

Total Matches 0

Playing Style Left Hander

Winning Percentage NaN%

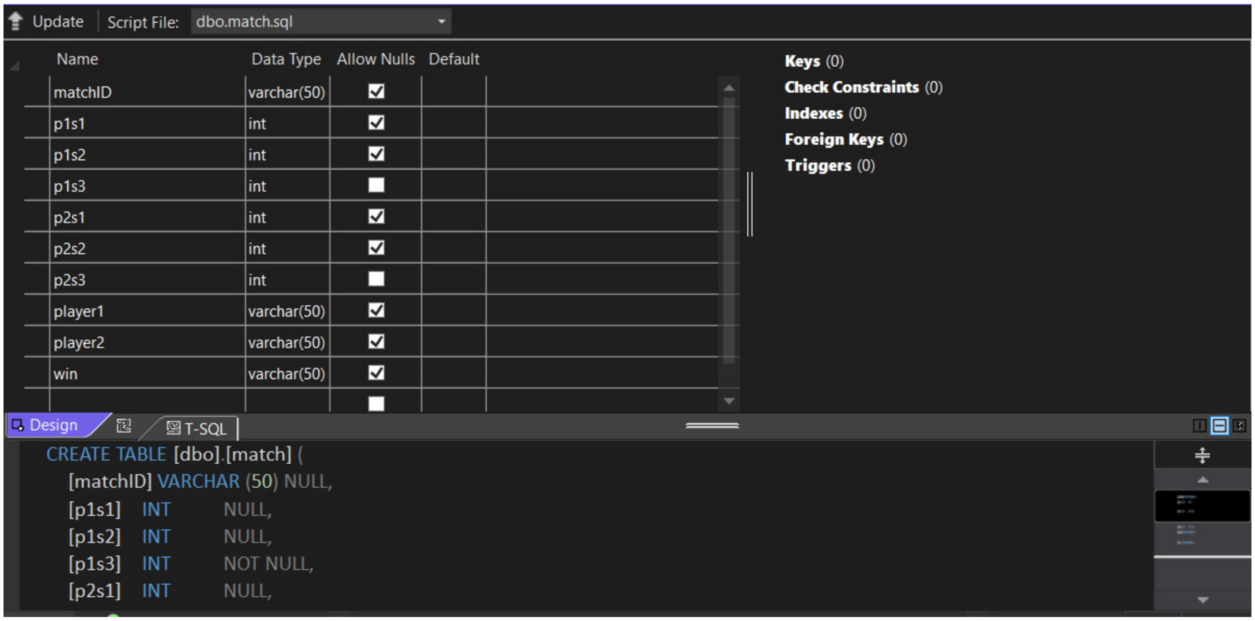
Update Player Details


Shuttler - Update Player Details
×
Enter the Player Student Id to Update
Name
Update
Telephone
Update
Email
Update

Databases

Database name: playerdb

Table name: match



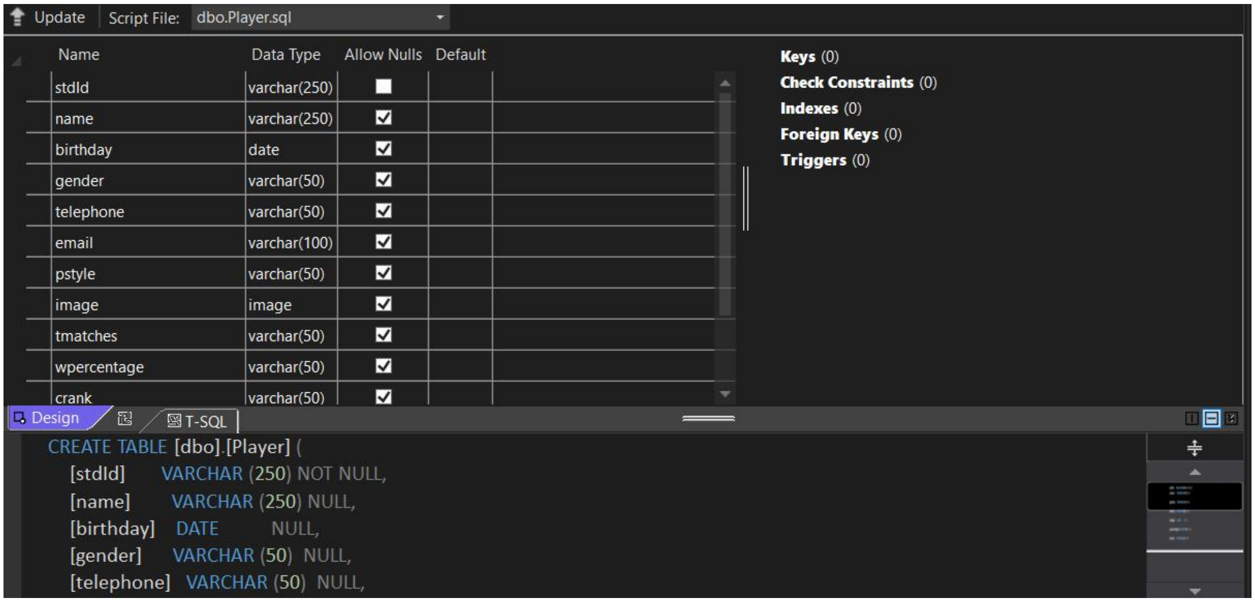
Script File: dbo.match.sql

Name	Data Type	Allow Nulls	Default
matchID	varchar(50)	☑	
p1s1	int	☑	
p1s2	int	☑	
p1s3	int	□	
p2s1	int	☑	
p2s2	int	☑	
p2s3	int	□	
player1	varchar(50)	☑	
player2	varchar(50)	☑	
win	varchar(50)	☑	
		□	

Keys (0)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[match] (
    [matchID] VARCHAR (50) NULL,
    [p1s1]    INT          NULL,
    [p1s2]    INT          NULL,
    [p1s3]    INT          NOT NULL,
    [p2s1]    INT          NULL,
```

Table name: Player



Script File: dbo.Player.sql

Name	Data Type	Allow Nulls	Default
stdId	varchar(250)	□	
name	varchar(250)	☑	
birthday	date	☑	
gender	varchar(50)	☑	
telephone	varchar(50)	☑	
email	varchar(100)	☑	
pstyle	varchar(50)	☑	
image	image	☑	
tmatches	varchar(50)	☑	
wpercentage	varchar(50)	☑	
crank	varchar(50)	☑	

Keys (0)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[Player] (
    [stdId]     VARCHAR (250) NOT NULL,
    [name]      VARCHAR (250) NULL,
    [birthday]  DATE          NULL,
    [gender]    VARCHAR (50)  NULL,
    [telephone] VARCHAR (50)  NULL,
```

8. Conclusion

This project has been challenging, but we persevered, therefore the project has repaid us with experience in more ways than one. This is an example of a common real-world situation. Our understanding of database design has grown because of the right procedures used to compile database design final reports. Scheduling a project and sticking to that timeline instills a strong sense of time management. Finally, our teamwork understanding has grown, and our confidence in handling real-life projects has grown significantly.

9. System Limitations

- Cannot get the highest rank and current rank from this system.
  The reason is mentioned below.

  There is a player called Player X who plays 99 out of 100 matches and wins. Then his winning percentage is 99/100. There is another player called Player Y who plays today and wins the match. Then his rank will be 1. But practically the rank of Player X should be 1. Also, considering the last few matches, the rank of Player Y is not 1 but Player X. Much of this is due to the complexity of the badminton scoring algorithm and the fact that we must wait at least some time until the end of a tournament to give the players' ranks.

- The players' match summary is unavailable.

- If the admin's computer has to be formatted, the system will lose data. So the data back upping is unavailable.

10. Contributors

Student ID	Student Name	Workload
23069	R.H.M.A.D. Herath	• The backend of the Player Search option • The backend of Update/Edit and Delete functions • System Testing part
23752	A.S.T. Umayanga	• Concept of Shuttler • The backend of Enter Player Details option
22043	P.K.W. Sapumohotti	• The backend of the Scoreboard window
22014	K.W.A.T. Dasun	• The frontend of the Scoreboard window
22980	D.S.W. Arachchi	• The frontend of Enter Player Details option
23164	T.N. Gunarathna	• System Analyzing • Report writing
24131	M.K.D.H.T. Amarasinghe	• The frontend of the Search Player option
21965	R.M.D.H. Rathnayake	• The frontend of the Update/Edit and Delete functions