1. INTRODUCTION

1.1 PROBLEM DEFINITION

Managing the intra-college or university sports activity is a daunting task irrespective of whether the institution accommodates a small student body or a large population. Administrators of such a program need to manage not just the sports activities, but also the teams and athletes that participate in the various sports as well as maintain statistics that are related to the program. In addition, coordinating the scheduling of contests, facilities, and officials as well as manipulating the large amount of data in various logical formats becomes an overwhelming task. A typical intramural program

1.2 PROJECT DESCRIPTION

The **College Sports Management System (CSMS)** objective is to provide which manages the activity of many sports at a time. It also manages the registration process and announcement of the results. The users will consume less amount of time when compared to manual paper work through the automated system. The system will take care of all the servicing activity in a quick manner.Data storing is easier. It will be able to check any report at any time. Paper work and manual work is reduced. The system is user friendly and easy to use.

Modules of CSMS

Add sports

We can add new sports into the system so that we are able to retrieve it later during the registration process. The sports added would be viewed during the creation of a new intra-college or university tournament.

Add scoreboard

We add scoresboard so that the students can view it and the results of each match are announced here so that there will be only one platform for the results. This would reduce the chaos during the score announcement.

Add Tournament

Each tournament from a intra-college or a university can be added here. It later would help in the registration of any sports in that tournament. While adding a new tournament the system would show the set of sports that are entered into the system by the director of the sports so only those sports can be included into the tournament while creation.

Remove sports

This module will help in the removal any sports that the sports director thinks is not needed in the system. The removed sports would be not be shown anywhere in the system that includes during the addition of a new tournament.

Edit scoreboard

The added scoreboards would be updated here. This module helps in the updating of the scores in the scoreboard. Only the scoreboard which are added using the add scoreboard module would be present here and only these scoreboards can be updated. We won't be able to add a new scoreboard here.

Remove players

This module would remove each players after each round of the tournament. So that only the existing player will be present and the one that are not qualified for the next round would be removed using this module. This would give a clear picture of the qualified players. As well as the player form the college team I can also be removed.

Remove Tournament

After each tournament in the college or a university we should remove it, so that there won't be any confusion between different tournaments which is going to be held later on. This module would help in removing all the details of the deleted tournament.

Registration Individual

This module would help in registration of individual sports events held in the tournament. We selected the tournament which we want to be part of and the sports which we want to participate in and the player would add his name and the required details asked in the registration form. After all these process then we can click on the submit button and the student has registration for the tournament that they wish to participate.

Registration Group

This module would help in registration of group sports events held in the tournament. We selected the tournament which we want to be part of and the sports which we want to participate in and the set of players name would be added and the other required details asked in the registration form. After all these process then we can click on the submit button and the student has registration for the tournament that they wish to participate.

Payment

This module would help in the online payment. So that the students wouldn't have to stand in the queue or have a hard cash in hand in order to do any payment to the sports department. By using this module we are reducing a lot of paper work and we are giving the students the liberty of doing the payment from wherever they are.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

The existing system is more of a manual work, where the students have to walk up to the sports department and have to register for the events that they desire to participate, it's not just the registration process but even other activities such as the announcement of the result, the payment etc. This results in a lot of paper work and chance of the data's getting mixed up are high. In the existing System, students are not able to get proper information about the games conducted in various venues. The student needs to spend a lot of time to get the information about the game. The student should attend the venue to get information on the game which takes a lot of time. The information such as the qualified players list, the timing of the event, the score of the sports happening etc.

DISADVANTAGE OF EXISTING SYSTEM

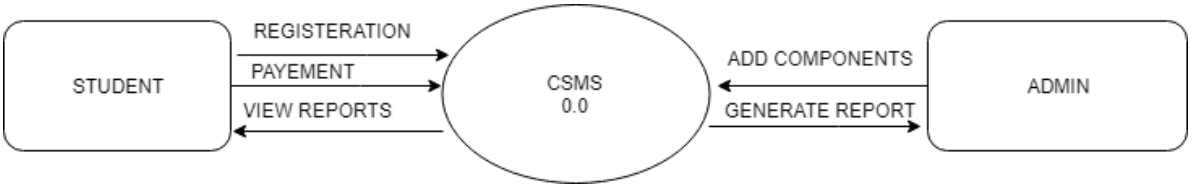
- Involves a lot of paper work
- Students have to walk a lot for the registration and other sports related activities
- Data getting corrupted is high
- Human error is common during an intra-college and university tournament.
- The students need to visit the venue to get all the information of the tournament and would have to wait for a long time.
- Manually recording all information with regard to all data and manually creating the contest schedules, coordinating facility usage, and hand-registering athletes and teams. The dissemination of information would require that documents be typed, photocopied and putted up in the notice board or common place where students can view.

2.2 PROPOSED SYSTEM

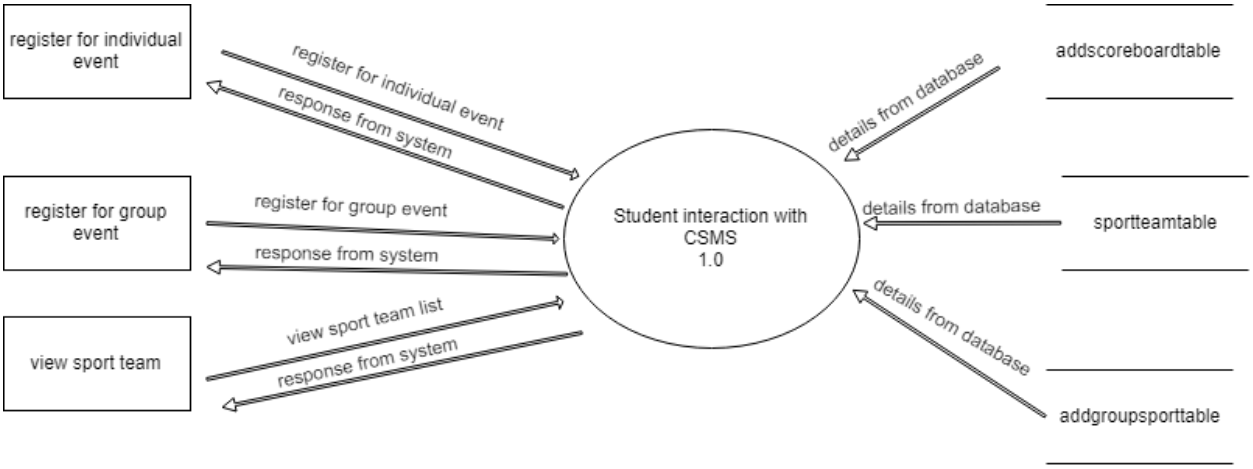
In the proposed College Sports Management System student can get all the information of various games and the venue. The student can get registered from anywhere and at any time. By using this system student can save a lot of time and effort. The student can easily get the information from anywhere. The proposed system is a web based system where the student will be an able access it using internet. It's more reliable and there is a log which is kept so that we are able to check history.

2.3 DATA FLOW DIAGRAM

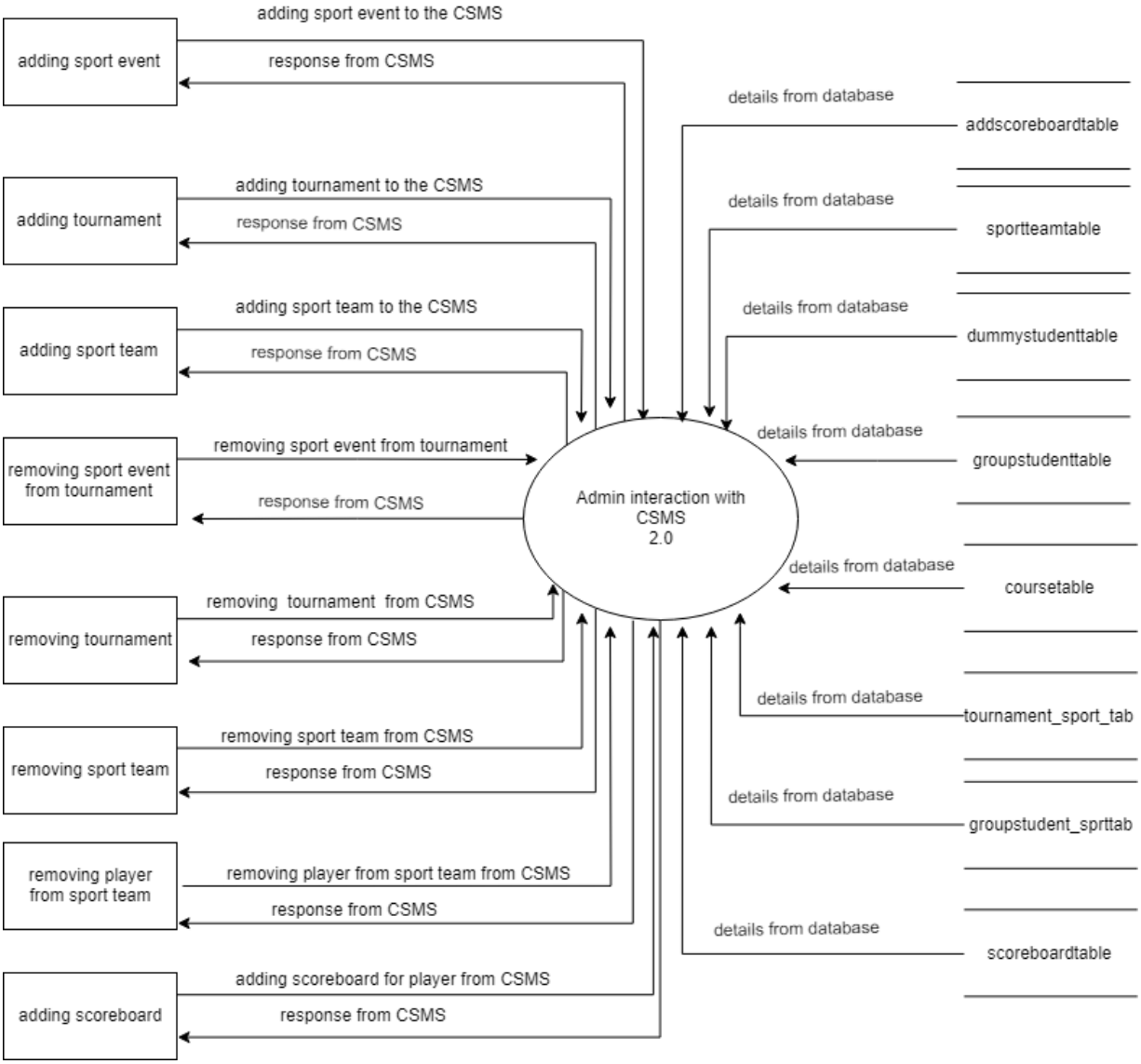
LEVEL 0



LEVEL 1



LEVEL 2


adding sport event to the CSMS
adding sport event
response from CSMS
details from database
addscoreboardtable
adding tournament to the CSMS
adding tournament
response from CSMS
details from database
sportteamtable
adding sport team to the CSMS
adding sport team
response from CSMS
details from database
dummystudenttable
removing sport event from tournament
removing sport event from tournament
response from CSMS
Admin interaction with CSMS 2.0
details from database
groupstudenttable
details from database
coursetable
removing tournament from CSMS
removing tournament
response from CSMS
details from database
tournament_sport_tab
removing sport team from CSMS
removing sport team
response from CSMS
details from database
groupstudent_sprttab
removing player from sport team from CSMS
removing player from sport team
response from CSMS
details from database
scoreboardtable
adding scoreboard for player from CSMS
adding scoreboard
response from CSMS

3. SYSTEM CONFIGURATION

3.1 HARDWARE CONFIGURATION

Processor : Pentium 4 or above

Hard disk : 80 GB

Ram : 512 MB SDRAM or above

3.2 SOFTWARE CONFIGURATION

Font – End : Microsoft ASP.NET

Code Behind : C# (C Sharp)

Back- End : SQL Server

Web server : Microsoft IIS

Operating System : Windows 8 or above

Framework : Bootstrap 4.0

Tools : Visual Studio 2008 or above,
Sql Management Studio

4. DETAILS OF SOFTWARE

4.1 OVERVIEW OF FRONTEND

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites. It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily.The language which we used for the project is C#.

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

C# makes it easy to develop software components through several innovative language constructs, including the following:

- Encapsulated method signatures called *delegates*, which enable type-safe event notifications.
- Properties, which serve as assessors for private member variables.
- Attributes, which provide declarative metadata about types at run time.
- Inline XML documentation comments.
- Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

4.2 OVERVIEW OF BACKEND

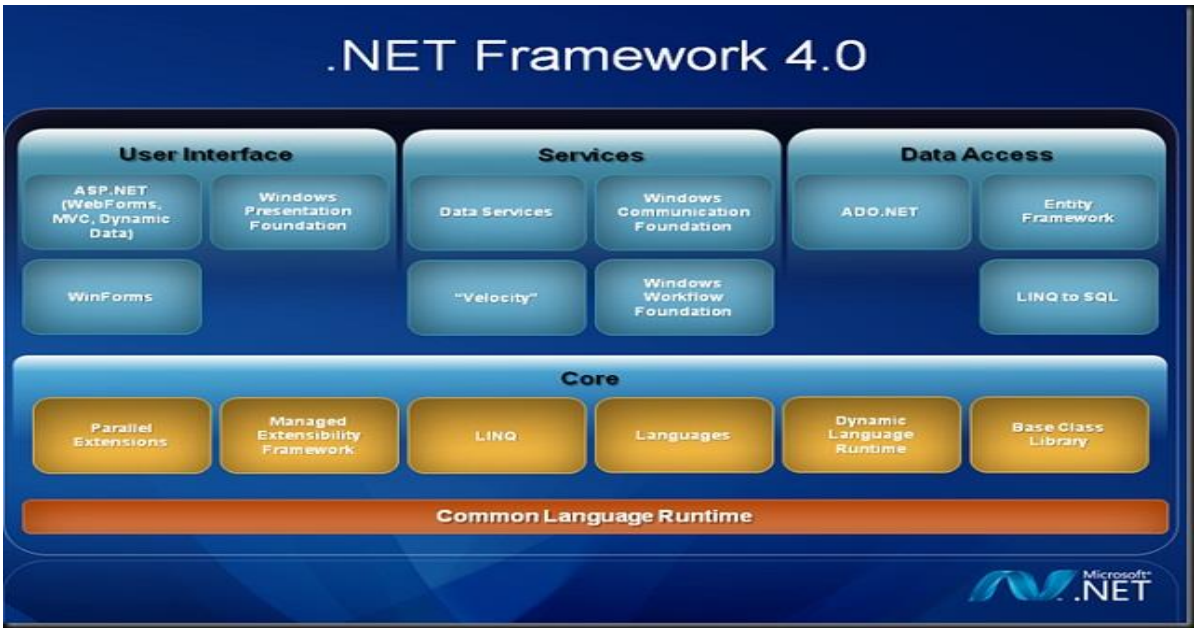
SQL SERVER

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

ABOUT THE PLATFORM

.NET Framework 4

Microsoft started development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released. The first version of .NET Framework was released on 13 February 2002, bringing managed code to Windows NT 4.0, 98, 2000, ME and XP.



The .NET Framework 4 works side by side with older Framework versions. Applications that are based on earlier versions of the Framework will continue to run on the version targeted by default.

The Microsoft .NET Framework 4 provides the following new features and improvements:

- Improvements in CLR and BCL
- Innovations in the Visual Basic and C# languages, for example statement lambdas, implicit line continuations, dynamic dispatch, and named/optional parameters.
- Improvements in ADO.NET

Enhancements to ASP.NET

- More control over HTML, element IDs and custom CSS that make it much easier to create standards-compliant and SEO-friendly web forms.
- Web forms support for new AJAX library improvements including built-in support for content delivery networks (CDNs).
- For a comprehensive list of enhancements to ASP.NET go here.
- Improvements in WPF
- Added support for Windows 7 multi-touch, ribbon controls, and taskbar extensibility features.
- Added support for Surface 2.0 SDK.

- New line-of-business controls including charting control, smart edit, data grid, and others that improve the experience for developers who build data centric applications.
- Improvements in performance and scalability.
- Visual improvements in text clarity, layout pixel snapping, localization, and interoperability.

Supported Operating Systems: Windows 7; Windows Server 2003; Windows Server 2008; Windows Vista.

.NET Framework 4 can be installed on the following operating systems:

- Windows XP SP3
- Windows Server 2003 SP2
- Windows Vista SP1 or later
- Windows Server 2008 (not supported on Server Core Role)
- Windows 7
- Windows Server 2008 R2 (not supported on Server Core Role)

Supported Architectures:

- x86
- x64

ia64 (some features are not supported on ia64 for example, WPF

SET OF TOOLS USED (FRONTEND)

- **LABEL**
  This control is used to display textual information on the web forms. It is mainly used to create caption for the other controls like: textbox.

  Example: < asp:LabelID="Label1" runat="server" Text="Label" ></asp:Label>

- **TEXTBOX**
  This is an input control which is used to take user input. To create **TextBox** either we can write code or use the drag and drop facility of visual studio IDE.

  Example: < asp:TextBoxID="TextBox1" runat="server" ></asp:TextBox>

- **DROPDOWNLIST**
  The DropDownList is a web server control which is used to create an HTML Select component. It allows us to select an option from the dropdown list. It can contain any number of items

  Example: <asp:DropDownList ID="DropDownList1" runat="server" >
  <asp:ListItem>NewYork</asp:ListItem>
  <asp:ListItem>Bangalore</asp:ListItem>

</asp:DropDownList>

- **CHECKBOX**
  It is used to get multiple inputs from the user. It allows user to select choices from the set of choices.It takes user input in yes or no format. It is useful when we want multiple choices from the user.To create CheckBox we can drag it from the toolbox in visual studio.

- **GRIDVIEW**
  The GridView control displays the values of a data source in a table. Each column represents a field, while each row represents a record.

- **PANEL**
  The Panel control works as a container for other controls on the page. It controls the appearance and visibility of the controls it contains. It also allows generating controls programmatically.

  Example : <asp:Panel ID= "Panel1" runat = "server">
  </asp:Panel>

- **MULTIVIEW**
  MultiView and View controls allow you to divide the content of a page into different groups, displaying only one group at a time. Each View control manages one group of content and all the View controls are held together in a MultiView control.

  Examples : <asp:MultView ID= "MultiView1" runat= "server">
  </asp:MultiView>

- **VIEW**
  View is a user interface. View displays data from the model to the user and also enables them to modify the data.

- **SQLDATASOURCE**
  It represents a connection to an ADO.NET data provider that returns SQL data, including data sources accessible via OLEDB and ODBC.

- **BUTTON**
  This control is used to perform events. It is also used to submit client request to the server. To create **Button** either we can write code or use the drag and drop facility of visual studio IDE.

  Example :
  < asp:ButtonID="Button1" runat="server" Text="Submit" BorderStyle="Solid" ToolTip= "Submit"/>

- **IMAGE BUTTON**
  ImageButton control in ASP.NET is just like a Button control. The ImageButton control is used to display a clickable image. Events of ImageButton are the same as Button control

- **LINK BUTTON**
  It is a server web control that acts as a hyperlink. It is used to display a hyperlink-style button control on the web page. ASP.NET provides a tag to create LinkButton and has following syntax.

- **RADIO BUTTON**
  It is an input control which is used to takes input from the user. It allows user to select a choice from the group of choices. To create Radio Button we can drag it from the toolbox of visual studio.

COMMON SQL QUERY USED

- select tournamentid,tname from addtournamenttable;

- select sportid,sportname,max_players from sporteventtable JOIN tournament_sport_table ON sporteventtable.sportid=tournament_sport_table.sid where tid=@tid;

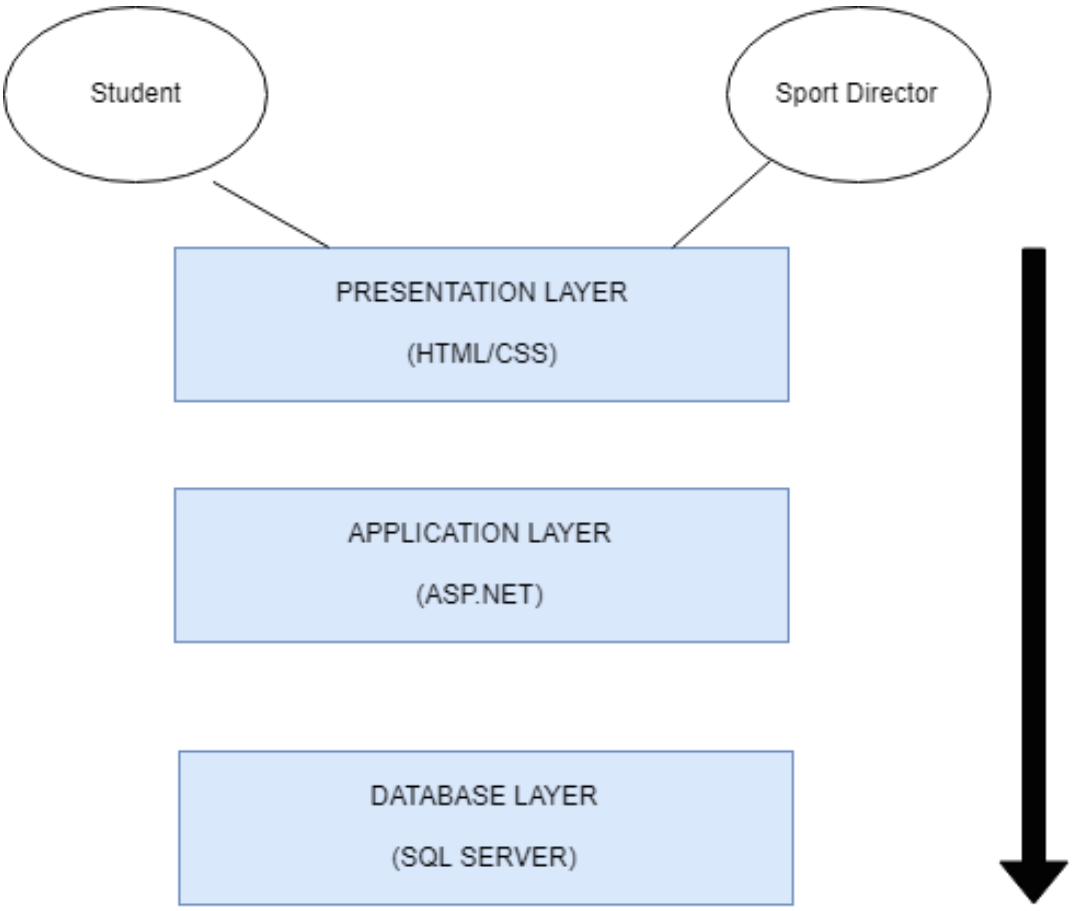
- insert into sporteventtable values (@spn,@spd,@max);

- select max_players from sporteventtable where sportid=@id;

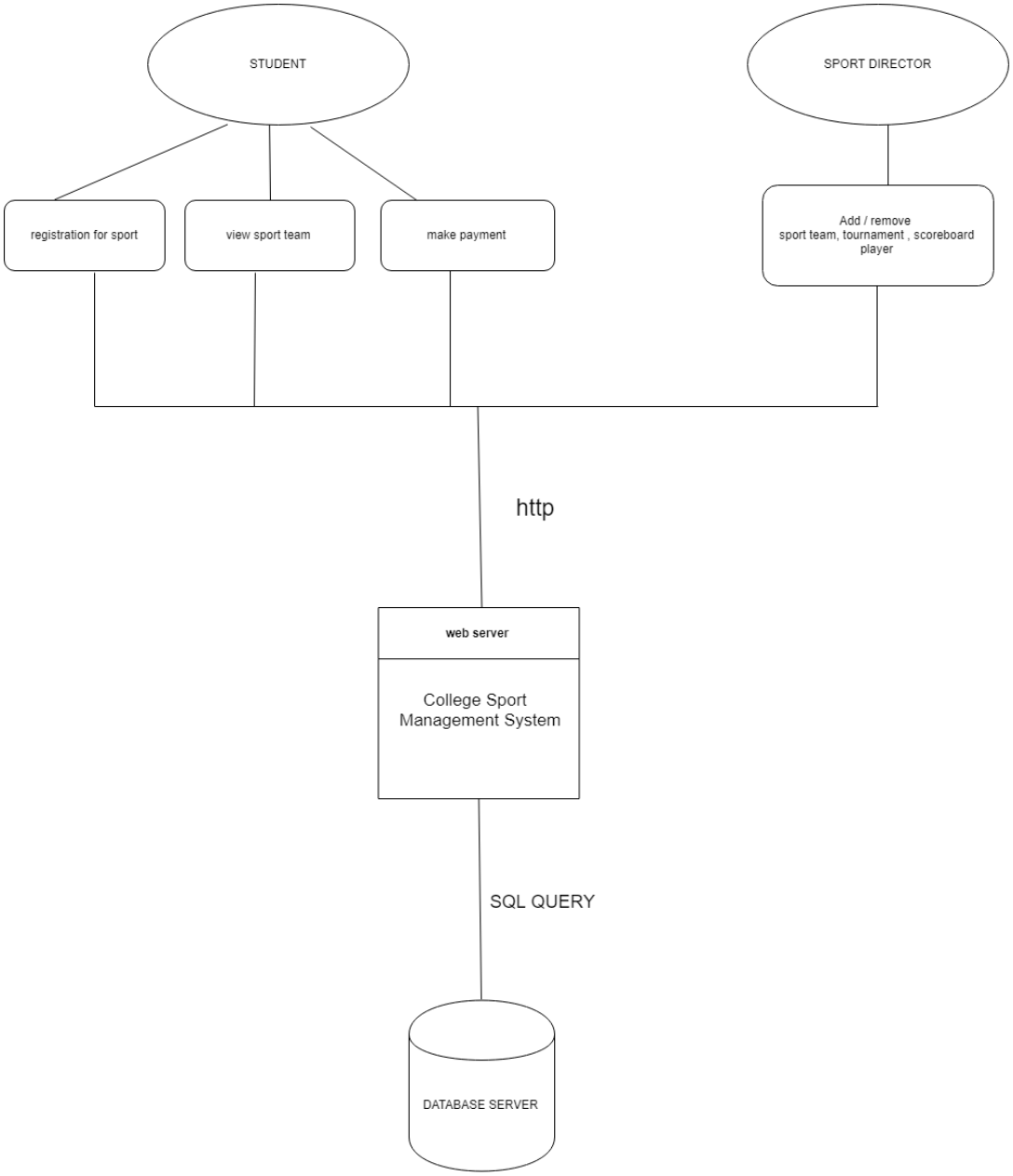
- Update scoreboardtable set score=@score where studentid=@studentid and tid=@tid and sid=@sid

5. SYSTEM DESIGN

5.1 ARCHITECTURAL DESIGN

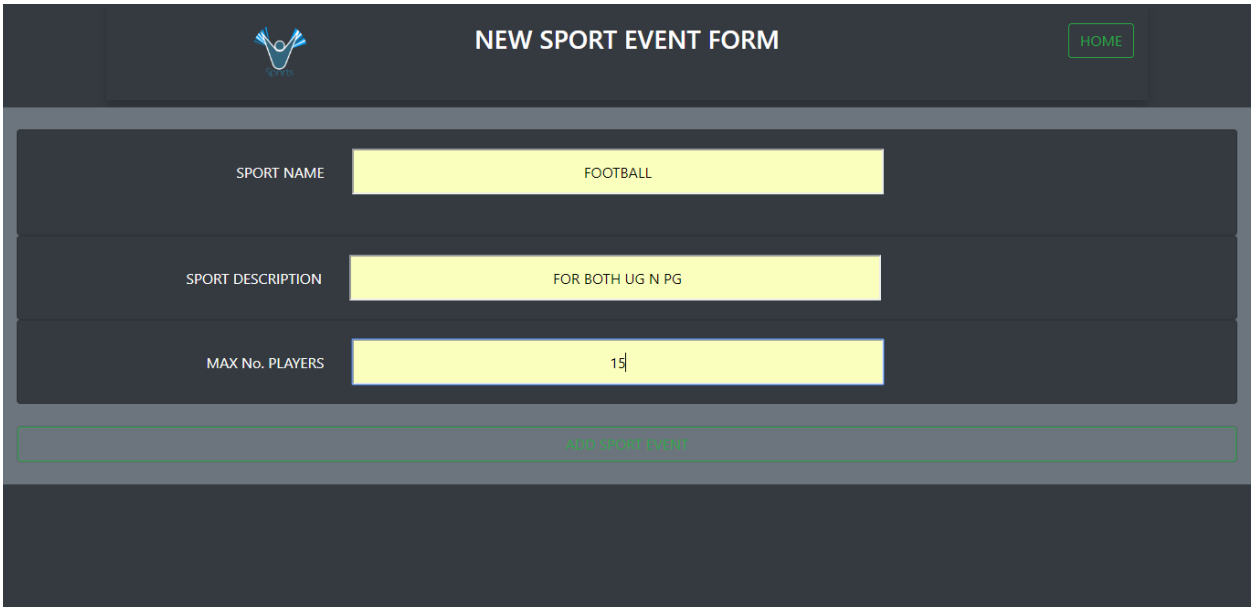


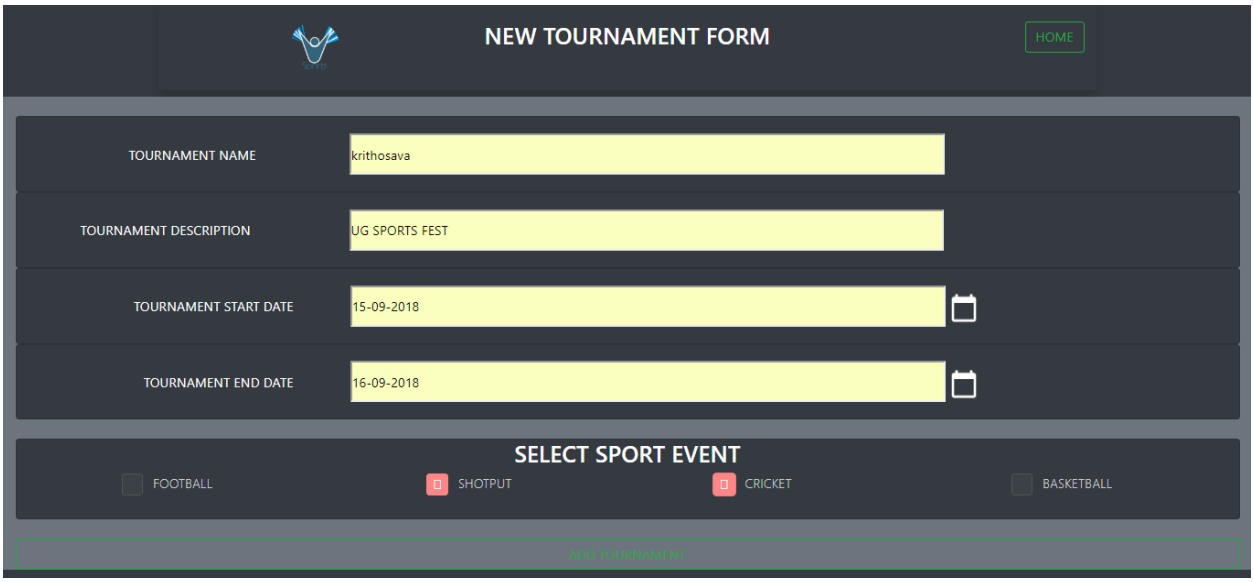
Overview of the different layer in the CSMS system

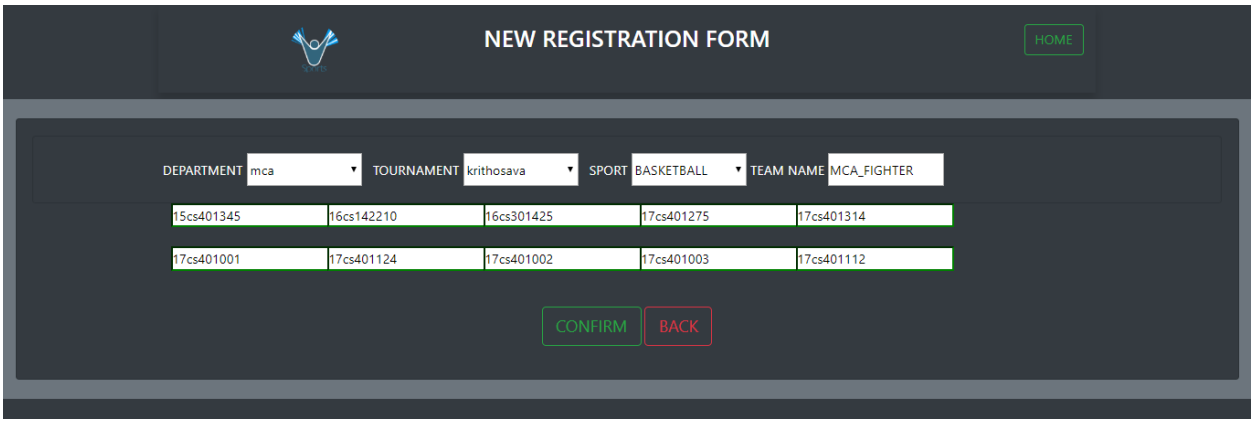


The architectural diagram of the system with the users and the relation of them to the system, along with the connection of the webserver and the database.

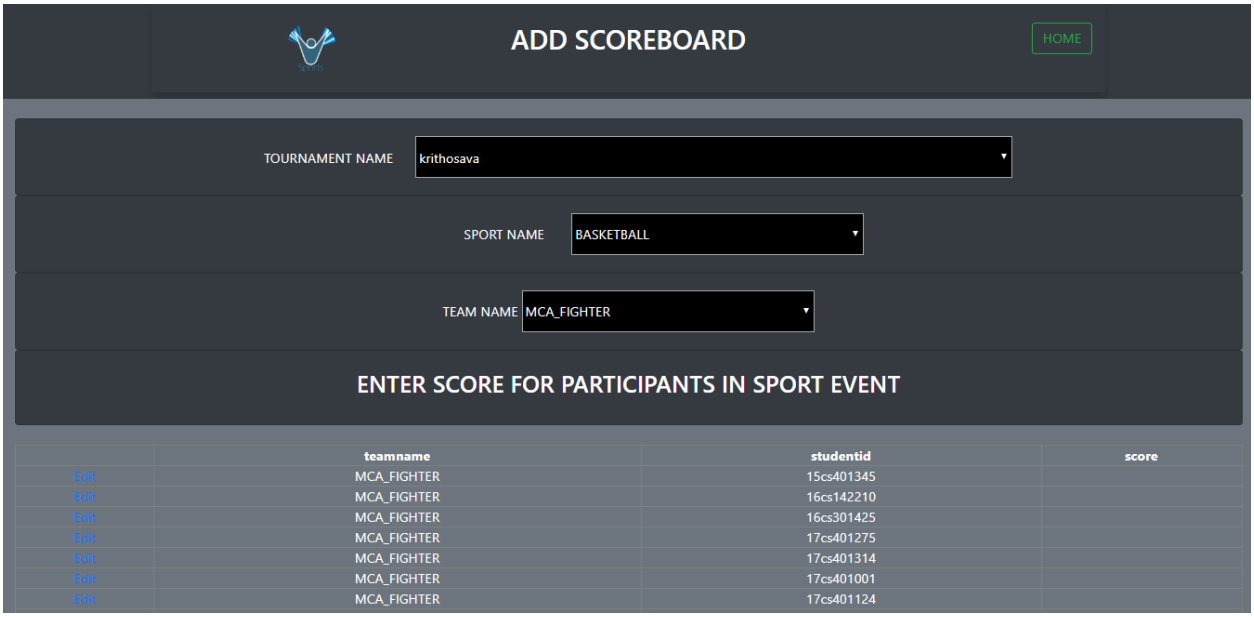
5.2 INPUT DESIGN

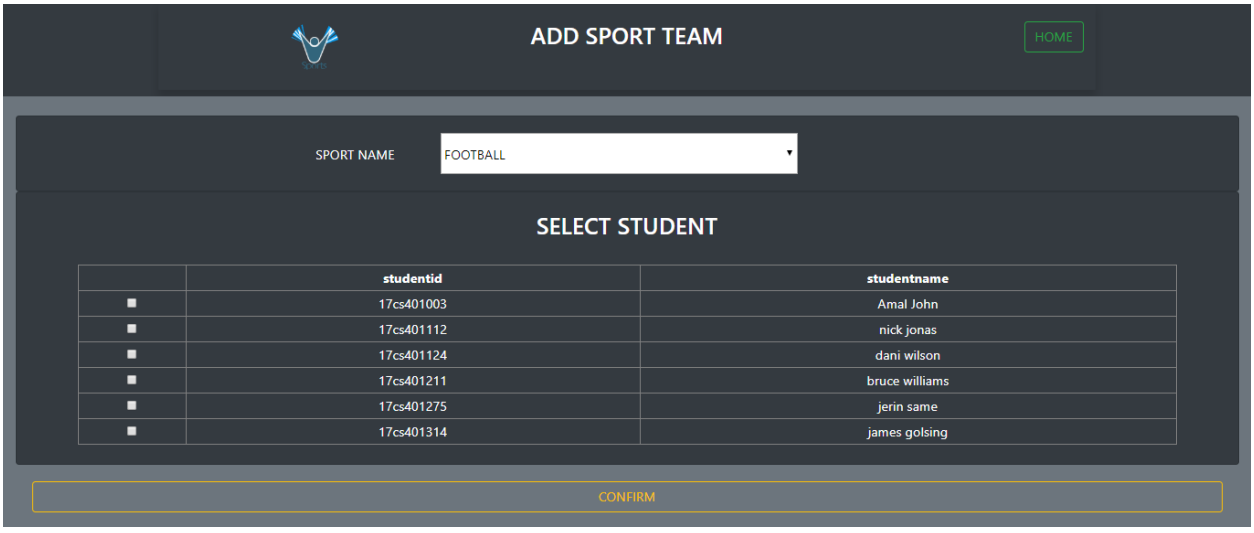


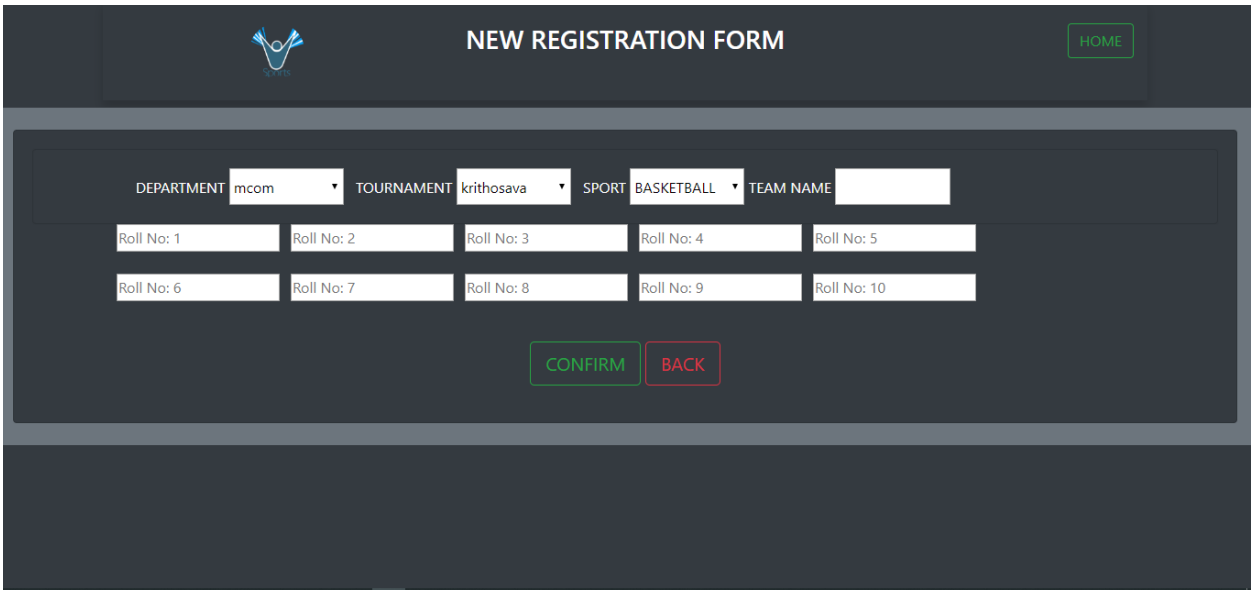



NEW REGISTRATION FORM
HOME
DEPARTMENT mca
TOURNAMENT krithosava
SPORT BASKETBALL
TEAM NAME MCA_FIGHTER
15cs401345
16cs142210
16cs301425
17cs401275
17cs401314
17cs401001
17cs401124
17cs401002
17cs401003
17cs401112
CONFIRM
BACK

5.3 OUTPUT DESIGN







5.4 DATABASE DESIGN

1. **Course table**

Update | Script File: dbo.coursetable.sql

	Name	Data Type	Allow Nulls	Default
🔑	courseid	varchar(50)	☐	
	coursename	text	☐	
	course_desc	text	☐	
			☐	

2**. Dummy students table**

Update | Script File: dbo.dummystudenttable.sql

	Name	Data Type	Allow Nulls	Default
🔑	studentid	varchar(50)	☐	
	studentname	text	☐	
	mobile	numeric(10,0)	☐	
	dob	varchar(50)	☐	
	courseid	varchar(50)	☐	
	section	text	☐	
			☐	

3. **Group sports details**

Update | Script File: dbo.groupsportdetail.sql

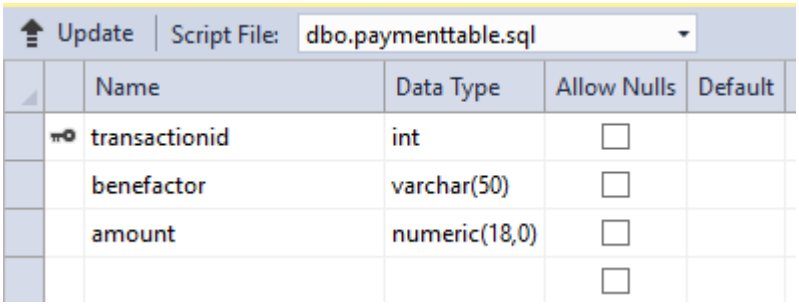
	Name	Data Type	Allow Nulls	Default
🔑	token	int	☐	
	sportid	int	☐	
	tournamentid	int	☐	
	teamname	varchar(50)	☐	
	courseid	varchar(50)	☐	
			☐	

4. **Group sports students list**

Update | Script File: dbo.groupsporteventstudentlist.sc

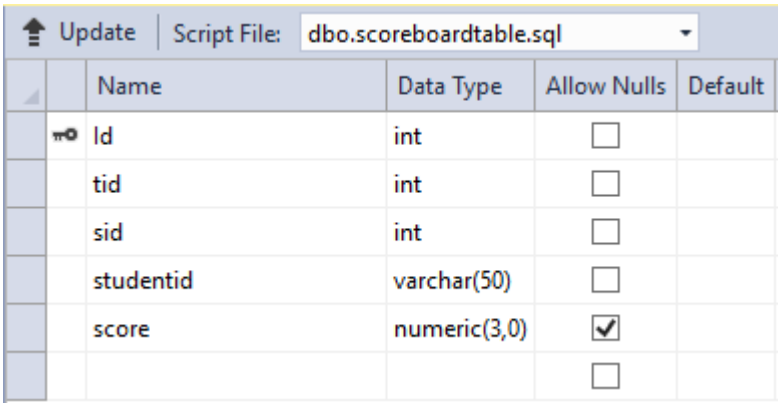
	Name	Data Type	Allow Nulls	Default
	token	int	☐	
	studentid	varchar(50)	☐	
	score	int	☑	
			☐	

5. **Payment table**



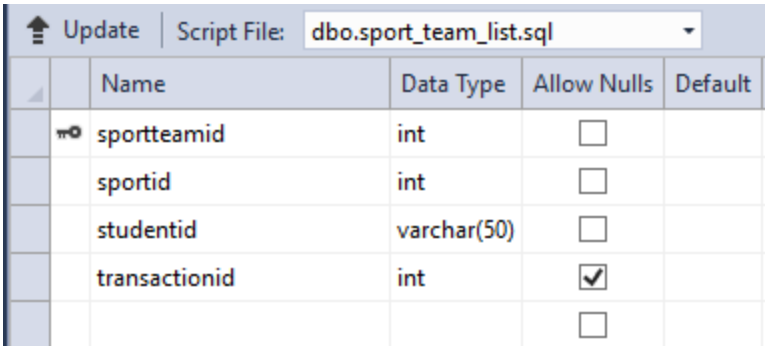
Name	Data Type	Allow Nulls	Default
transactionid	int	☐	
benefactor	varchar(50)	☐	
amount	numeric(18,0)	☐	
		☐	

6. **Scoreboard table**



Name	Data Type	Allow Nulls	Default
Id	int	☐	
tid	int	☐	
sid	int	☐	
studentid	varchar(50)	☐	
score	numeric(3,0)	☑	
		☐	

7**. Sports team table**



Name	Data Type	Allow Nulls	Default
sportteamid	int	☐	
sportid	int	☐	
studentid	varchar(50)	☐	
transactionid	int	☑	
		☐	

8. **Sports table**

Update | Script File: dbo.sporteventtable.sql

Name	Data Type	Allow Nulls	Default
sportid	int	☐	
sportname	varchar(50)	☐	
sportdesc	text	☐	
max_players	int	☐	
		☐	

9. **Tournament table**

Update | Script File: dbo.tournament_sport_table.sql

Name	Data Type	Allow Nulls	Default
tid	int	☐	
sid	int	☐	
		☐	

10. **Creating tournament table**

Update | Script File: dbo.addtournamenttable.sql

Name	Data Type	Allow Nulls	Default
tournamentid	int	☐	
tname	varchar(50)	☐	
tstartdate	varchar(50)	☐	
tenddate	varchar(50)	☐	
tdesc	text	☐	
		☐	

6. SOURCE CODE

Score boards (back end)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace webapp_sportmanagement
{
   public partial class addscoreboard : System.Web.UI.Page
   {
      SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\DANIEL\\source\\repos\\S
port Mangement\\webapp sportmanagement\\App_Data\\newdb.mdf;Integrated Security=True");
      protected void Page_Load(object sender, EventArgs e)
      {
         if(!IsPostBack)
         {
            //BindData();
            con.Open();
            SqlCommand cmd = new SqlCommand("select tournamentid,tname from
addtournamenttable", con);
            SqlDataReader dr = cmd.ExecuteReader();
            DDL1.DataSource = dr;
            DDL1.Items.Clear();
            DDL1.DataTextField = "tname";
            DDL1.DataValueField = "tournamentid";
            DDL1.DataBind();
            con.Close();
            DDL1.Items.Insert(0, new ListItem("--Select tournament--", "0"));
         }

      }

      protected void DDL1_SelectedIndexChanged(object sender, EventArgs e)
      {
         con.Open();
```

```
        SqlCommand cmd = new SqlCommand("select sportid,sportname,max_players from
sporteventtable JOIN tournament_sport_table ON
sporteventtable.sportid=tournament_sport_table.sid where tid=@tid", con);
        cmd.Parameters.AddWithValue("@tid", DDL1.SelectedValue.ToString());
        SqlDataReader dr = cmd.ExecuteReader();

        DDL2.DataSource = dr;
        DDL2.Items.Clear();
        DDL2.DataTextField = "sportname";
        DDL2.DataValueField = "sportid";
        DDL2.DataBind();
        con.Close();
        DDL2.Items.Insert(0, new ListItem("--please select sport--", "0"));
    }
    protected void DDL2_SelectedIndexChanged(object sender, EventArgs e)
    {

        con.Open();
        SqlCommand cm = new SqlCommand("select max_players from sporteventtable where
sportid=@id ", con);
        cm.Parameters.AddWithValue("@id", DDL2.SelectedValue.ToString());
        int max = (int)cm.ExecuteScalar();
        con.Close();
        if (max == 1)
        {
            GridView2.Visible = false;
            GridView1.Visible = true;
            team.Visible = false;
            teamlist.Visible = false;
            BindData();

        }
        else
        {
            GridView1.Visible = false;
            team.Visible = true;
            teamlist.Visible = true;
            con.Open();
            SqlCommand cmd = new SqlCommand("select groupsportdetail.teamname,
groupsportdetail.token from groupsportdetail where groupsportdetail.tournamentid=@tid and
groupsportdetail.sportid=@sid", con);
            cmd.Parameters.AddWithValue("@tid", DDL1.SelectedValue.ToString());
            cmd.Parameters.AddWithValue("@sid", DDL2.SelectedValue.ToString());
            SqlDataReader dr = cmd.ExecuteReader();

            teamlist.DataSource = dr;
```

```
            teamlist.Items.Clear();
            teamlist.DataTextField = "teamname";
            teamlist.DataValueField = "token";
            teamlist.DataBind();
            con.Close();
            teamlist.Items.Insert(0, new ListItem("--please select a team--", "0"));
        }



    }
    protected void teamlist_SelectedIndexChanged(object sender, EventArgs e)
    {
        BindDatagroup();
        GridView1.Visible = false;
        GridView2.Visible = true;
    }
    public void BindDatagroup()
    {
        con.Open();
        SqlDataAdapter sda = new SqlDataAdapter("select groupsportdetail.teamname,
groupsporteventstudentlist.studentid, groupsporteventstudentlist.score from groupsportdetail join
groupsporteventstudentlist on groupsportdetail.token=groupsporteventstudentlist.token where
groupsportdetail.tournamentid='" + DDL1.SelectedValue.ToString() + "' and
groupsportdetail.sportid='" + DDL2.SelectedValue.ToString() + "' and groupsportdetail.token='"
+ teamlist.SelectedValue.ToString() + "'", con);

        DataTable dt = new DataTable();
        sda.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            GridView2.DataSource = dt;
            GridView2.DataBind();
        }
        con.Close();
    }
    public void BindData()
    {
        con.Open();
        SqlDataAdapter sda = new SqlDataAdapter("select
dummystudenttable.studentid,dummystudenttable.studentname,scoreboardtable.score from
dummystudenttable JOIN scoreboardtable ON
scoreboardtable.studentid=dummystudenttable.studentid where tid='" +
DDL1.SelectedValue.ToString() + "' and sid='" + DDL2.SelectedValue.ToString() + "'", con);
        DataTable dt = new DataTable();
        sda.Fill(dt);
```

```
        if (dt.Rows.Count > 0)
        {
            GridView1.DataSource = dt;
            GridView1.DataBind();
        }
        con.Close();
    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        GridView1.EditIndex = e.NewEditIndex;
        BindData();
    }

    protected void GridView1_RowUpdating(object sender,
System.Web.UI.WebControls.GridViewUpdateEventArgs e)
    {
        GridViewRow row = GridView1.Rows[e.RowIndex];

        string studentid = (row.Cells[1].Controls[0] as TextBox).Text;
        string score = (row.Cells[3].Controls[0] as TextBox).Text;
        con.Open();
        SqlCommand cmd = new SqlCommand("Update scoreboardtable set score=@score
where studentid=@studentid and tid=@tid and sid=@sid", con);
        cmd.Parameters.AddWithValue("@tid", DDL1.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@sid", DDL2.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@score", score);
        cmd.Parameters.AddWithValue("@studentid", studentid);
        cmd.ExecuteNonQuery();
        con.Close();
        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView1.EditIndex = -1;
        //Call ShowData method for displaying updated data
        BindData();
    }
    protected void GridView1_RowCancelingEdit(object sender,
System.Web.UI.WebControls.GridViewCancelEditEventArgs e)
    {
        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView1.EditIndex = -1;
        BindData();
    }



    protected void GridView2_RowEditing(object sender, GridViewEditEventArgs e)
    {
```

```
        GridView2.EditIndex = e.NewEditIndex;
        BindDatagroup();
    }

    protected void GridView2_RowUpdating(object sender,
System.Web.UI.WebControls.GridViewUpdateEventArgs e)
    {
        GridViewRow row = GridView2.Rows[e.RowIndex];

        string studentid = (row.Cells[2].Controls[0] as TextBox).Text;
        string score = (row.Cells[3].Controls[0] as TextBox).Text;
        con.Open();
        SqlCommand cmd = new SqlCommand("Update groupsporteventstudentlist set
score=@score where studentid=@studentid and token=@token", con);
        cmd.Parameters.AddWithValue("@token", teamlist.SelectedValue.ToString());

        cmd.Parameters.AddWithValue("@score", score);
        cmd.Parameters.AddWithValue("@studentid", studentid);
        cmd.ExecuteNonQuery();
        con.Close();


        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView2.EditIndex = -1;
        //Call ShowData method for displaying updated data
        BindDatagroup();
    }
    protected void GridView2_RowCancelingEdit(object sender,
System.Web.UI.WebControls.GridViewCancelEditEventArgs e)
    {
        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView2.EditIndex = -1;
        BindDatagroup();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("admin_dashboard.aspx");
    }
  }

}
```

Add sports event (back end)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace webapp_sportmanagement
{
   public partial class addsportevent : System.Web.UI.Page
   {
      SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\DANIEL\\source\\repos\\S
port Mangement\\webapp sportmanagement\\App_Data\\newdb.mdf;Integrated Security=True");
      protected void Page_Load(object sender, EventArgs e)
      {
         ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         String sportname = TB1.Text;
         String sportdesc = TB2.Text;
         String maxplyr = TB3.Text;
         SqlCommand cmd = new SqlCommand("insert into sporteventtable values
(@spn,@spd,@max)", con);
         cmd.Parameters.AddWithValue("@spn", sportname);
         cmd.Parameters.AddWithValue("@spd", sportdesc);
         cmd.Parameters.AddWithValue("@max", Convert.ToInt32(maxplyr));
         try
         {
            con.Open();
            cmd.ExecuteNonQuery();
            String strmsg = "event" +sportname + " successfully created!!";

            con.Close();
           // ScriptManager.RegisterStartupScript(this, GetType(), "alertMessage", "alert('"
+sportname + "');", true);

            Response.Redirect("admin_dashboard.aspx");
```

```
        }
        catch (Exception ex) { Response.Write(ex); }
      }
   }
}
```

Registration (front end)

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="registeration_form1.aspx.cs"
Inherits="webapp_sportmanagement.registeration_form1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>REGISTER FOR EVENTS</title>
   <meta charset="utf-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1" />
   <link rel="stylesheet" href="assets/css/main.css" type="text/css" />
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" />
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
   <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
</head>
<body class="bg-dark text-white">
   <form id="form1" runat="server">
      <div class="container shadow mb-2 bg-transparent">
         <br />
         <div class="row">
            <div class="col-sm-4">
               <img src="images/csmslogo.png" class="mx-auto d-block align-content-sm-center
rounded-circle " alt="CSMS" height="60" width="60" />
            </div>
            <div class="col-sm-4">
               <h3 class="text-center mx-auto d-block align-middle  ">NEW REGISTRATION
FORM</h3>
            </div>
            <div class="col-sm-4">
               <a href="index.aspx" id="home" class="btn btn-outline-success float-sm-right
">HOME</a>
            </div>
         </div>
```

```
        <br />
      </div>
      <div class="container-fluid bg-secondary">
        <br />
        <div class="card-group-vertical text-white">
          <div class="card bg-dark ">
            <div class="card-body text-center ">
              <asp:MultiView ID="MultiView1" runat="server">
                <asp:View ID="View1" runat="server">

                  <asp:Button ID="individual" class="btn btn-lg  btn-success" runat="server"
Text="INDIVIDUAL EVENT" OnClick="individual_Click" />
                  <asp:Button ID="Group" class="btn btn-lg btn-success " runat="server"
Text="GROUP EVENT" OnClick="Group_Click" />
                </asp:View>
                <asp:View ID="View2" runat="server">
                  <div class="card bg-dark ">
                    <div class="card-body text-center ">
                      <asp:Label ID="Label1" runat="server" Text="TOURNAMENT
NAME" ></asp:Label>
                      <asp:DropDownList ID="DDL1" Width="30%" AutoPostBack="True"
Height="50px" runat="server" BackColor="white" ForeColor="black"
OnSelectedIndexChanged="DDL1_SelectedIndexChanged">
                      </asp:DropDownList>
                      <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
class="float-right" runat="server" ErrorMessage="TOURNAMENT NAME REQUIRED *"
InitialValue="0" ControlToValidate="DDL1" ForeColor="Red"></asp:RequiredFieldValidator>
                    </div>
                  </div>
                  <div class="card bg-dark ">
                    <div class="card-body text-center">
                      <asp:Label ID="Label11" runat="server" Text="STUDENT
ID"></asp:Label>
                      <asp:TextBox ID="TextBox11" runat="server" Width="30%"
Height="50" AutoPostBack="True"
OnTextChanged="Textbox11_onTextchanged"></asp:TextBox>
                    </div>
                  </div>
                  <div class="btn-group  bg-outline-success">
                    <div class="btn btn-lg w-100 text-center">
                      <asp:Label ID="Label2" runat="server" Text="STUDENT
NAME"></asp:Label>
                      <asp:TextBox ID="TextBox2" Width="30%" Height="50"
runat="server" Enabled="False"></asp:TextBox>
                    </div>
                    <div class="btn  btn-lg w-100 text-center">
```

```
                        <asp:Label ID="Label3" runat="server" Text="DATE OF
BIRTH"></asp:Label>
                        <asp:TextBox ID="TextBox3" Width="40%" Height="50"
runat="server" Enabled="False"></asp:TextBox>
                    </div>
                </div>
                <div class="btn-group bg-outline-success">
                    <div class="btn btn-lg w-100 text-center">
                        <asp:Label ID="Label4" runat="server" Text="MOBILE
NUMBER"></asp:Label>
                        <asp:TextBox ID="TextBox4" Width="40%" Height="50"
runat="server" Enabled="False"></asp:TextBox>
                    </div>
                    <div class="btn btn-lg w-100 text-center">
                        <asp:Label ID="Label5" runat="server" Text="COURSE
ID"></asp:Label>
                        <asp:TextBox ID="TextBox5" Width="30%" Height="50"
runat="server" Enabled="False"></asp:TextBox>
                    </div>
                </div>
                <div class="card bg-dark text-center text-white">
                    <h3>SELECT SPORT EVENT</h3>
                    <br />
                    <asp:CheckBoxList ID="CheckBoxList1" css="mx-auto" runat="server"
RepeatDirection="Horizontal" DataSourceID="SqlDataSource1" DataTextField="sportname"
DataValueField="sportid">
                    </asp:CheckBoxList>
                    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>" ></asp:SqlDataSource>
                    <br />
                </div>
                <div class="card-body text-center ">
                    <asp:Button ID="Button1" class="btn btn-lg btn-outline-success"
runat="server" Text="CONFIRM" OnClick="Button1_Click" />
                    <asp:Button ID="back" class="btn btn-lg btn-outline-danger"
runat="server" Text="BACK" OnClick="back_Click" CausesValidation="False" />
                </div>
            </asp:View>
            <asp:View ID="View3" runat="server">
                <div class="card bg-dark ">
                    <div class="card-body text-center ">
                        <asp:Label ID="deptname" runat="server" Text="DEPARTMENT"
></asp:Label>
                        <asp:DropDownList ID="Deptlist" Width="10%"
AutoPostBack="True" Height="40px" runat="server" BackColor="white" ForeColor="black"
OnSelectedIndexChanged="Deptlist_SelectedIndexChanged" >
```

```
                        </asp:DropDownList>
                    <span> </span>
                        <asp:Label ID="tourname" runat="server" Text="TOURNAMENT"
></asp:Label>
                        <asp:DropDownList ID="tourlist" Width="10%"
AutoPostBack="True" Height="40px" runat="server" BackColor="white" ForeColor="black"
OnSelectedIndexChanged="tourlist_SelectedIndexChanged" >
                        </asp:DropDownList>

                    <span> </span>

                        <asp:Label ID="sportn" runat="server" Text="SPORT" ></asp:Label>
                        <asp:DropDownList ID="toursportlist" Width="10%"
AutoPostBack="True" Height="40px" runat="server" BackColor="white" ForeColor="black"
OnSelectedIndexChanged="toursportlist_SelectedIndexChanged" >
                        </asp:DropDownList>

                        <asp:Label ID="teamnamelabel" runat="server" Text="TEAM NAME"
></asp:Label>
                        <asp:TextBox ID="teamname" Width="10%" Height="40px"
runat="server"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="teamval3" class="float-right"
runat="server" ErrorMessage="TEAM NAME REQUIRED *" ControlToValidate="teamname"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </div>
                </div>
                <asp:MultiView ID="MultiView2" runat="server">
                    <asp:View ID="none" runat="server"></asp:View>
                    <asp:View ID="FOOTBALL" runat="server">
                        <div class="container">
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p1" AutoPostBack="true" placeholder="Roll
No: 1" OnTextChanged="p1_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p2" AutoPostBack="true" placeholder="Roll
No: 2" OnTextChanged="p2_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p3" AutoPostBack="true" placeholder="Roll
No: 3" runat="server" OnTextChanged="p3_onTextchanged"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p4" AutoPostBack="true" placeholder="Roll
No: 4" OnTextChanged="p4_onTextchanged" runat="server"></asp:TextBox>
```

```
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p5" AutoPostBack="true" placeholder="Roll
No: 5" OnTextChanged="p5_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p6" AutoPostBack="true" placeholder="Roll
No: 6" OnTextChanged="p6_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p7" AutoPostBack="true" placeholder="Roll
No: 7" OnTextChanged="p7_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p8" AutoPostBack="true" placeholder="Roll
No: 8" OnTextChanged="p8_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p9" AutoPostBack="true" placeholder="Roll
No: 9" OnTextChanged="p9_onTextchanged" runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p10" AutoPostBack="true"
placeholder="Roll No: 10" OnTextChanged="p10_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p11" AutoPostBack="true"
placeholder="Roll No: 11" OnTextChanged="p11_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="p12" AutoPostBack="true"
placeholder="Roll No: 12" OnTextChanged="p12_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-4  ">
                                    <asp:TextBox ID="p13" AutoPostBack="true"
placeholder="Roll No: 13" OnTextChanged="p13_onTextchanged"
runat="server"></asp:TextBox>
```

```
                                </div>
                                <div class="col-sm-4  ">
                                    <asp:TextBox ID="p14" AutoPostBack="true"
placeholder="Roll No: 14" OnTextChanged="p14_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-4  ">
                                    <asp:TextBox ID="p15" AutoPostBack="true"
placeholder="Roll No: 15" OnTextChanged="p15_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                            </div>
                        </div>
                        <div class="card-body text-center ">
                            <asp:Button ID="grpbtnconfirmp1" class="btn btn-lg btn-outline-
success" runat="server" Text="CONFIRM" OnClick="grpbtnconfirmp1_Click" />
                            <asp:Button ID="grpbtnbackp1" class="btn btn-lg btn-outline-
danger" runat="server" Text="BACK" OnClick="grpbtnbackp1_Click"
CausesValidation="False" />

                        </div>
                    </asp:View>
                    <asp:View ID="CRICKET" runat="server">
                        <div class="container">
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc1" AutoPostBack="true"
placeholder="Roll No: 1" OnTextChanged="pc1_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc2" AutoPostBack="true"
placeholder="Roll No: 2" OnTextChanged="pc2_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc3" AutoPostBack="true"
placeholder="Roll No: 3" runat="server"
OnTextChanged="pc3_onTextchanged"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc4" AutoPostBack="true"
placeholder="Roll No: 4" OnTextChanged="pc4_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
```

```
                                    <asp:TextBox ID="pc5" AutoPostBack="true"
placeholder="Roll No: 5" OnTextChanged="pc5_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc6" AutoPostBack="true"
placeholder="Roll No: 6" OnTextChanged="pc6_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc7" AutoPostBack="true"
placeholder="Roll No: 7" OnTextChanged="pc7_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc8" AutoPostBack="true"
placeholder="Roll No: 8" OnTextChanged="pc8_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc9" AutoPostBack="true"
placeholder="Roll No: 9" OnTextChanged="pc9_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc10" AutoPostBack="true"
placeholder="Roll No: 10" OnTextChanged="pc10_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc11" AutoPostBack="true"
placeholder="Roll No: 11" OnTextChanged="pc11_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pc12" AutoPostBack="true"
placeholder="Roll No: 12" OnTextChanged="pc12_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-4  ">
```

```
                                    <asp:TextBox ID="pc13" AutoPostBack="true"
placeholder="Roll No: 13" OnTextChanged="pc13_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-4  ">
                                    <asp:TextBox ID="pc14" AutoPostBack="true"
placeholder="Roll No: 14" OnTextChanged="pc14_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-4  ">
                                    <asp:TextBox ID="pc15" AutoPostBack="true"
placeholder="Roll No: 15" OnTextChanged="pc15_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                            </div>
                        </div>
                        <div class="card-body text-center ">
                            <asp:Button ID="grpbtnconfirmpc1" class="btn btn-lg btn-outline-
success" runat="server" Text="CONFIRM" OnClick="grpbtnconfirmpc1_Click" />
                            <asp:Button ID="grpbtnbackpc1" class="btn btn-lg btn-outline-
danger" runat="server" Text="BACK" OnClick="grpbtnbackpc1_Click"
CausesValidation="False" />

                        </div>
                    </asp:View>
                    <asp:View ID="BASKETBALL" runat="server">
                        <div class="container">
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb1" AutoPostBack="true"
placeholder="Roll No: 1" OnTextChanged="pb1_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb2" AutoPostBack="true"
placeholder="Roll No: 2" OnTextChanged="pb2_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb3" AutoPostBack="true"
placeholder="Roll No: 3" runat="server"
OnTextChanged="pb3_onTextchanged"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
```

```
                                    <asp:TextBox ID="pb4" AutoPostBack="true"
placeholder="Roll No: 4" OnTextChanged="pb4_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb5" AutoPostBack="true"
placeholder="Roll No: 5" OnTextChanged="pb5_onTextchanged"
runat="server"></asp:TextBox>
                                </div>

                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb6" AutoPostBack="true"
placeholder="Roll No: 6" OnTextChanged="pb6_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb7" AutoPostBack="true"
placeholder="Roll No: 7" OnTextChanged="pb7_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb8" AutoPostBack="true"
placeholder="Roll No: 8" OnTextChanged="pb8_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb9" AutoPostBack="true"
placeholder="Roll No: 9" OnTextChanged="pb9_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pb10" AutoPostBack="true"
placeholder="Roll No: 10" OnTextChanged="pb10_onTextchanged"
runat="server"></asp:TextBox>
                                </div>

                            </div>
                            <br />

                        </div>
                        <div class="card-body text-center ">
                            <asp:Button ID="grpbtnconfirmpb1" class="btn btn-lg btn-outline-
success" runat="server" Text="CONFIRM" OnClick="grpbtnconfirmpb1_Click" />
```

```
                        <asp:Button ID="grpbtnbackpb1" class="btn btn-lg btn-outline-
danger" runat="server" Text="BACK" OnClick="grpbtnbackpb1_Click"
CausesValidation="False" />

                    </div>
                </asp:View>
                <asp:View ID="VOLLEYBALL" runat="server">
                    <div class="container">
                        <div class="row bg-dark ">
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv1" AutoPostBack="true"
placeholder="Roll No: 1" OnTextChanged="pv1_onTextchanged"
runat="server"></asp:TextBox>
                            </div>
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv2" AutoPostBack="true"
placeholder="Roll No: 2" OnTextChanged="pv2_onTextchanged"
runat="server"></asp:TextBox>
                            </div>
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv3" AutoPostBack="true"
placeholder="Roll No: 3" runat="server"
OnTextChanged="pv3_onTextchanged"></asp:TextBox>
                            </div>
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv4" AutoPostBack="true"
placeholder="Roll No: 4" OnTextChanged="pv4_onTextchanged"
runat="server"></asp:TextBox>
                            </div>
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv5" AutoPostBack="true"
placeholder="Roll No: 5" OnTextChanged="pv5_onTextchanged"
runat="server"></asp:TextBox>
                            </div>

                        </div>
                        <br />
                        <div class="row bg-dark ">
                            <div class="col-sm-2  ">
                                <asp:TextBox ID="pv6" AutoPostBack="true"
placeholder="Roll No: 6" OnTextChanged="pv6_onTextchanged"
runat="server"></asp:TextBox>
                            </div>
                            <div class="col-sm-2  ">
```

```
                                    <asp:TextBox ID="pv7" AutoPostBack="true"
placeholder="Roll No: 7" OnTextChanged="pv7_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pv8" AutoPostBack="true"
placeholder="Roll No: 8" OnTextChanged="pv8_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pv9" AutoPostBack="true"
placeholder="Roll No: 9" OnTextChanged="pv9_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pv10" AutoPostBack="true"
placeholder="Roll No: 10" OnTextChanged="pv10_onTextchanged"
runat="server"></asp:TextBox>
                                </div>

                            </div>
                            <br />

                        </div>
                        <div class="card-body text-center ">
                            <asp:Button ID="grpbtnconfirmpv1" class="btn btn-lg btn-outline-
success" runat="server" Text="CONFIRM" OnClick="grpbtnconfirmpv1_Click" />
                            <asp:Button ID="grpbtnbackpv1" class="btn btn-lg btn-outline-
danger" runat="server" Text="BACK" OnClick="grpbtnbackpv1_Click"
CausesValidation="False" />

                        </div>
                    </asp:View>
                    <asp:View ID="THROWBALL" runat="server">
                        <div class="container">
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt1" AutoPostBack="true"
placeholder="Roll No: 1" OnTextChanged="pt1_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt2" AutoPostBack="true"
placeholder="Roll No: 2" OnTextChanged="pt2_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
```

```
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt3" AutoPostBack="true"
placeholder="Roll No: 3" runat="server"
OnTextChanged="pt3_onTextchanged"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt4" AutoPostBack="true"
placeholder="Roll No: 4" OnTextChanged="pt4_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt5" AutoPostBack="true"
placeholder="Roll No: 5" OnTextChanged="pt5_onTextchanged"
runat="server"></asp:TextBox>
                                </div>

                            </div>
                            <br />
                            <div class="row bg-dark ">
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt6" AutoPostBack="true"
placeholder="Roll No: 6" OnTextChanged="pt6_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt7" AutoPostBack="true"
placeholder="Roll No: 7" OnTextChanged="pt7_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt8" AutoPostBack="true"
placeholder="Roll No: 8" OnTextChanged="pt8_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt9" AutoPostBack="true"
placeholder="Roll No: 9" OnTextChanged="pt9_onTextchanged"
runat="server"></asp:TextBox>
                                </div>
                                <div class="col-sm-2  ">
                                    <asp:TextBox ID="pt10" AutoPostBack="true"
placeholder="Roll No: 10" OnTextChanged="pt10_onTextchanged"
runat="server"></asp:TextBox>
                                </div>

                            </div>
```

```
                                <br />

                            </div>
                            <div class="card-body text-center ">
                                <asp:Button ID="grpbtnconfirmpt1" class="btn btn-lg btn-outline-
success" runat="server" Text="CONFIRM" OnClick="grpbtnconfirmpt1_Click" />
                                <asp:Button ID="grpbtnbackpt1" class="btn btn-lg btn-outline-
danger" runat="server" Text="BACK" OnClick="grpbtnbackpt1_Click"
CausesValidation="False" />

                            </div>
                        </asp:View>

                    </asp:MultiView>

                </asp:View>
            </asp:MultiView>
        </div>
    </div>

    <br />
  </div>
 </div>

 </form>
</body>
</html>
```

Registration Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.ComponentModel;
using System.Drawing;
using System.Text;
using System.Configuration;
namespace webapp_sportmanagement
{
  public partial class registeration_form1 : System.Web.UI.Page
  {
```

```
    SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\DANIEL\\source\\repos\\S
port Mangement\\webapp sportmanagement\\App_Data\\newdb.mdf;Integrated Security=True");
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            MultiView1.ActiveViewIndex = 0;
        }
    }
    //check if rollno exist for individual player
    protected void Textbox11_onTextchanged(object sender, EventArgs e)
    {
        try
        {
            String roll = TextBox11.Text;
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from dummystudenttable where
studentid=@roll", con);
            cmd.Parameters.AddWithValue("@roll", roll);
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    TextBox2.Text = dr["studentname"].ToString();
                    TextBox3.Text = dr["dob"].ToString();
                    TextBox4.Text = dr["mobile"].ToString();
                    TextBox5.Text = dr["courseid"].ToString();
                    TextBox2.Enabled = false;
                    TextBox3.Enabled = false;
                    TextBox4.Enabled = false;
                    TextBox5.Enabled = false;
                }
            }
            else
            {
                TextBox2.Text = "";
                TextBox3.Text = "";
                TextBox4.Text = "";
                TextBox5.Text = "";
                TextBox2.Enabled = true;
                TextBox4.Enabled = true;
                TextBox3.Enabled = true;
                TextBox5.Enabled = true;
            }
```

```
            con.Close();
        }
        catch (Exception ex) { Response.Write(ex); }
    }
    protected void back_Click(object sender, EventArgs e)
    {
        MultiView1.ActiveViewIndex = 0;
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        String tid = DDL1.SelectedValue.ToString();
        String roll = TextBox11.Text;
        int count = 0;
        foreach (ListItem list in CheckBoxList1.Items)
        {
            if (list.Selected == true)
            {
                count++;
                con.Open();
                SqlDataAdapter sd = new SqlDataAdapter("select studentid from scoreboardtable",
con);
                DataSet ds = new DataSet();
                sd.Fill(ds, "id");
                List<string> stid = new List<string>();
                foreach (DataRow row in ds.Tables["id"].Rows)
                {
                    stid.Add(row["studentid"].ToString());
                }
                con.Close();
                Boolean check = stid.Contains(TextBox11.Text);
                if (check == false)
                {
                    try
                    {
                      con.Open();
                        SqlCommand cmd2 = new SqlCommand("insert into scoreboardtable
(tid,studentid,sid) values (@tid,@studentid,@sid)", con);
                        cmd2.Parameters.AddWithValue("@tid", tid);
                        cmd2.Parameters.AddWithValue("@studentid", roll);
                        cmd2.Parameters.AddWithValue("@sid", list.Value);
                        cmd2.ExecuteNonQuery();
                        con.Close();
                        ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Succesfully registered');", true);
                    }
                    catch (Exception ew) { Response.Write(ew); }
```

```
                }
                else
                {
                    ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Already
exist for an individual event');", true);
                }

            }



        }
        if (count == 0)
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Please select
a sport event to continue');", true);
        }
    }
    //selecting sport for dropdown for individual event
    protected void DDL1_SelectedIndexChanged(object sender, EventArgs e)
    {
        //  SqlDataSource1.SelectParameters.Clear();
        SqlDataSource1.SelectCommand = "SELECT sportid, sportname FROM sporteventtable
JOIN tournament_sport_table on sporteventtable.sportid=tournament_sport_table.sid WHERE
sporteventtable.max_players=1 and tournament_sport_table.tid='" +
DDL1.SelectedValue.ToString() + "'";
        // SqlDataSource1.SelectParameters.Add("td", DDL1.SelectedValue.ToString());

    }
    //individual sport view
    protected void individual_Click(object sender, EventArgs e)
    {
        MultiView1.ActiveViewIndex = 1;
        ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;

        con.Open();
        SqlCommand cmd = new SqlCommand("select tournamentid,tname from
addtournamenttable", con);
        SqlDataReader dr = cmd.ExecuteReader();
        DDL1.DataSource = dr;
        DDL1.DataTextField = "tname";
        DDL1.DataValueField = "tournamentid";
        DDL1.DataBind();
        con.Close();
        DDL1.Items.Insert(0, new ListItem("--Select tournament--", "0"));
    }
    //group sports view
```

```
    protected void Group_Click(object sender, EventArgs e)
    {
        MultiView1.ActiveViewIndex = 2;
        MultiView2.ActiveViewIndex = 0;
        ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
        con.Open();
        SqlCommand cmd1 = new SqlCommand("select courseid,coursename from coursetable",
con);
        SqlDataReader dr1 = cmd1.ExecuteReader();
        Deptlist.DataSource = dr1;
        Deptlist.DataTextField = "coursename";
        Deptlist.DataValueField = "courseid";
        Deptlist.DataBind();
        con.Close();
        Deptlist.Items.Insert(0, new ListItem("--Select department--", "0"));









    }

    //selecting dept
    protected void Deptlist_SelectedIndexChanged(object sender, EventArgs e)
    {

        ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
        con.Open();
        SqlCommand cmd = new SqlCommand("select tournamentid,tname from
addtournamenttable", con);
        SqlDataReader dr = cmd.ExecuteReader();
        tourlist.DataSource = dr;
        tourlist.DataTextField = "tname";
        tourlist.DataValueField = "tournamentid";
        tourlist.DataBind();
        con.Close();
        tourlist.Items.Insert(0, new ListItem("--Select tournament--", "0"));
    }
    //selecting tournament for group sport
    protected void tourlist_SelectedIndexChanged(object sender, EventArgs e)
    {
        ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
        con.Open();
        SqlCommand cmd1 = new SqlCommand("select sid,sportname,max_players from
tournament_sport_table join sporteventtable on sporteventtable.sportid =
```

```
tournament_sport_table.sid and sporteventtable.max_players > 1  where
tournament_sport_table.tid=@tid", con);
        cmd1.Parameters.AddWithValue("@tid", tourlist.SelectedValue.ToString());
        SqlDataReader dr1 = cmd1.ExecuteReader();
        toursportlist.DataSource = dr1;
        toursportlist.DataTextField = "sportname";
        toursportlist.DataValueField = "sid";
        toursportlist.DataBind();
        con.Close();
        toursportlist.Items.Insert(0, new ListItem("--Select sport--", "0"));
    }
    //selecting sport for group sport
    protected void toursportlist_SelectedIndexChanged(object sender, EventArgs e)
    {
        ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
        Response.Write(toursportlist.SelectedItem.ToString());
        switch (toursportlist.SelectedItem.ToString())
        {
            case "FOOTBALL":
                MultiView2.ActiveViewIndex = 1;



                break;
            case "CRICKET":
                MultiView2.ActiveViewIndex = 2;
                break;
            case "BASKETBALL":
                MultiView2.ActiveViewIndex = 3;
                break;
            case "VOLLEYBALL":
                MultiView2.ActiveViewIndex = 4;
                break;
            case "THROWBALL":
                MultiView2.ActiveViewIndex = 5;
                break;
            case "HOCKEY":
                MultiView2.ActiveViewIndex = 6;
                break;

        }


    }
```

```
    //footbal grp event starts....
    protected void grpbtnbackp1_Click(object sender, EventArgs e)
    {
        MultiView1.ActiveViewIndex = 0;
    }
    //checking if rollno exist for group registeristion
    protected void p1_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@roll", p1.Text);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p1.BorderColor = Color.Green;
        }
        else
        {
            p1.BorderColor = Color.Red;

        }
    }
    protected void p2_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p2.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p2.BorderColor = Color.Green;
        }
        else
        {
            p2.BorderColor = Color.Red;

        }
    }
    protected void p3_onTextchanged(object sender, EventArgs e)
```

```
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p3.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p3.BorderColor = Color.Green;
        }
        else
        {
            p3.BorderColor = Color.Red;
        }
    }
    protected void p4_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p4.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p4.BorderColor = Color.Green;
        }
        else
        {
            p4.BorderColor = Color.Red;
        }
    }
    protected void p5_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p5.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
```

```
            p5.BorderColor = Color.Green;
        }
        else
        {
            p5.BorderColor = Color.Red;
        }
    }
    protected void p6_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p6.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p6.BorderColor = Color.Green;
        }
        else
        {
            p6.BorderColor = Color.Red;
        }
    }
    protected void p7_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p7.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p7.BorderColor = Color.Green;
        }
        else
        {
            p7.BorderColor = Color.Red;
        }
    }
    protected void p8_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p8.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p8.BorderColor = Color.Green;
        }
        else
        {
           p8.BorderColor = Color.Red;
        }
    }
    protected void p9_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p9.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p9.BorderColor = Color.Green;
        }
        else
        {
           p9.BorderColor = Color.Red;
        }
    }
    protected void p10_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p10.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p10.BorderColor = Color.Green;
        }
```

```
        else
        {
            p10.BorderColor = Color.Red;
        }
    }
    protected void p11_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p11.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p11.BorderColor = Color.Green;
        }
        else
        {
            p11.BorderColor = Color.Red;
        }
    }
    protected void p12_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p12.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            p12.BorderColor = Color.Green;
        }
        else
        {
            p12.BorderColor = Color.Red;
        }
    }
    protected void p13_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
```

```
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p13.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p13.BorderColor = Color.Green;
        }
        else
        {
           p13.BorderColor = Color.Red;
        }
    }
    protected void p14_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", p14.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p14.BorderColor = Color.Green;
        }
        else
        {
           p14.BorderColor = Color.Red;
        }
    }
    protected void p15_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll", con);
        cmd.Parameters.AddWithValue("@roll", p15.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           p15.BorderColor = Color.Green;
        }
        else
        {
           p15.BorderColor = Color.Red;
```

```
        }
    }
    //grp sport confirmation
    protected void grpbtnconfirmp1_Click(object sender, EventArgs e)
    {
        if ((p1.BorderColor == Color.Green) && (p2.BorderColor == Color.Green) &&
(p3.BorderColor == Color.Green) && (p4.BorderColor == Color.Green) && (p5.BorderColor
== Color.Green) && (p6.BorderColor == Color.Green) && (p7.BorderColor == Color.Green)
&& (p8.BorderColor == Color.Green) && (p9.BorderColor == Color.Green) &&
(p10.BorderColor == Color.Green) && (p11.BorderColor == Color.Green) &&
(p12.BorderColor == Color.Green) && (p13.BorderColor == Color.Green) &&
(p14.BorderColor == Color.Green) && (p15.BorderColor == Color.Green))
        {
            //check p1 unique
            if ((p1.Text != p2.Text) && (p1.Text != p3.Text) && (p1.Text != p4.Text) &&
(p1.Text != p5.Text) && (p1.Text != p6.Text) && (p1.Text != p7.Text) && (p1.Text !=
p8.Text) && (p1.Text != p9.Text) && (p1.Text != p10.Text) && (p1.Text != p11.Text) &&
(p1.Text != p12.Text) && (p1.Text != p13.Text) && (p1.Text != p14.Text) && (p1.Text !=
p15.Text))
            {
                //check p2 unique
                if ((p2.Text != p3.Text) && (p2.Text != p4.Text) && (p2.Text != p5.Text) &&
(p2.Text != p6.Text) && (p2.Text != p7.Text) && (p2.Text != p8.Text) && (p2.Text !=
p9.Text) && (p2.Text != p10.Text) && (p2.Text != p11.Text) && (p2.Text != p12.Text) &&
(p2.Text != p13.Text) && (p2.Text != p14.Text) && (p2.Text != p15.Text))
                {
                    //check p3 unique
                    if ((p3.Text != p4.Text) && (p3.Text != p5.Text) && (p3.Text != p6.Text) &&
(p3.Text != p7.Text) && (p3.Text != p8.Text) && (p3.Text != p9.Text) && (p3.Text !=
p10.Text) && (p3.Text != p11.Text) && (p3.Text != p12.Text) && (p3.Text != p13.Text) &&
(p3.Text != p14.Text) && (p3.Text != p15.Text))
                    {
                        //check p4 unique
                        if ((p4.Text != p5.Text) && (p4.Text != p6.Text) && (p4.Text != p7.Text) &&
(p4.Text != p8.Text) && (p4.Text != p9.Text) && (p4.Text != p10.Text) && (p4.Text !=
p11.Text) && (p4.Text != p12.Text) && (p4.Text != p13.Text) && (p4.Text != p14.Text) &&
(p4.Text != p15.Text))
                        {
                            //check p5 unique
                            if ((p5.Text != p6.Text) && (p5.Text != p7.Text) && (p5.Text != p8.Text)
&& (p5.Text != p9.Text) && (p5.Text != p10.Text) && (p5.Text != p11.Text) && (p5.Text !=
p12.Text) && (p5.Text != p13.Text) && (p5.Text != p14.Text) && (p5.Text != p15.Text))
                            {
                                //check p6 unique
```

```
                        if ((p6.Text != p7.Text) && (p6.Text != p8.Text) && (p6.Text != p9.Text)
&& (p6.Text != p10.Text) && (p6.Text != p11.Text) && (p6.Text != p12.Text) && (p6.Text !=
p13.Text) && (p6.Text != p14.Text) && (p6.Text != p15.Text))
                        {
                            //check p7 unique
                            if ((p7.Text != p8.Text) && (p7.Text != p9.Text) && (p7.Text !=
p10.Text) && (p7.Text != p11.Text) && (p7.Text != p12.Text) && (p7.Text != p13.Text) &&
(p7.Text != p14.Text) && (p7.Text != p15.Text))
                            {
                                //check p8 unique
                                if ((p8.Text != p9.Text) && (p8.Text != p10.Text) && (p8.Text !=
p11.Text) && (p8.Text != p12.Text) && (p8.Text != p13.Text) && (p8.Text != p14.Text) &&
(p8.Text != p15.Text))
                                {
                                    //check p9 unique
                                    if ((p9.Text != p10.Text) && (p9.Text != p11.Text) && (p9.Text
!= p12.Text) && (p9.Text != p13.Text) && (p9.Text != p14.Text) && (p9.Text != p15.Text))
                                    {
                                        //check p10 unique
                                        if ((p10.Text != p11.Text) && (p10.Text != p12.Text) &&
(p10.Text != p13.Text) && (p10.Text != p14.Text) && (p10.Text != p15.Text))
                                        {
                                            //check p11 unique
                                            if ((p11.Text != p12.Text) && (p11.Text != p13.Text) &&
(p11.Text != p14.Text) && (p11.Text != p15.Text))
                                            {
                                                //check p12 unique
                                                if ((p12.Text != p13.Text) && (p12.Text != p14.Text) &&
(p12.Text != p15.Text))
                                                {
                                                    //check p13 unique
                                                    if ((p13.Text != p14.Text) && (p13.Text != p15.Text))
                                                    {
                                                        //check p14 unique
                                                        if ((p14.Text != p15.Text))
                                                        {
                                                            int token;
                                                            //enter into db
                                                            con.Open();
                                                            SqlDataAdapter sd = new SqlDataAdapter("select
teamname from groupsportdetail", con);
                                                            DataSet ds = new DataSet();
                                                            sd.Fill(ds, "TNAME");
                                                            List<string> tname = new List<string>();
                                                            foreach (DataRow row in
ds.Tables["TNAME"].Rows)
```

```
                                        {
                                            tname.Add(row["teamname"].ToString());

                                        }
                                        con.Close();
                                        Boolean check = tname.Contains(teamname.Text);
                                        if (check == false)
                                        {
                                            con.Close();
                                            con.Open();
                                            SqlCommand cmd2 = new SqlCommand("insert
into groupsportdetail (teamname,sportid,courseid,tournamentid) values
(@teamname,@sportid,@courseid,@tournamentid)", con);
                                            cmd2.Parameters.AddWithValue("@teamname",
teamname.Text);
                                            cmd2.Parameters.AddWithValue("@sportid",
toursportlist.SelectedValue.ToString());
                                            cmd2.Parameters.AddWithValue("@courseid",
Deptlist.SelectedValue.ToString());

cmd2.Parameters.AddWithValue("@tournamentid", tourlist.SelectedValue.ToString());
                                            cmd2.ExecuteNonQuery();
                                            con.Close();

                                            con.Open();
                                            SqlCommand cmd = new SqlCommand("select
token from groupsportdetail where teamname=@teamname ", con);
                                            cmd.Parameters.AddWithValue("@teamname",
teamname.Text);
                                            token = (int)cmd.ExecuteScalar();
                                            con.Close();
                                            string[] value = { p1.Text, p2.Text, p3.Text,
p4.Text, p5.Text, p6.Text, p7.Text, p8.Text, p9.Text, p10.Text, p11.Text, p12.Text, p13.Text,
p14.Text, p15.Text };
                                            for (int i = 0; i < 15; i++)
                                            {
                                                con.Open();
                                                SqlCommand cmd1 = new
SqlCommand("insert into groupsporteventstudentlist (token,studentid) values
(@token,@studentid)", con);
                                                cmd1.Parameters.AddWithValue("@token",
token);

cmd1.Parameters.AddWithValue("@studentid", value[i]);
                                                cmd1.ExecuteNonQuery();
                                                con.Close();
```

```
                                                }
                                                ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Succesfully registered');", true);

                                            }
                                            else
                                            {
                                                ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('team name already taken');", true);
                                            }
                                        }
                                        else
                                        {
                                            ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                        }
                                    }
                                    else
                                    {
                                        ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                    }
                                }
                                else
                                {
                                    ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                }
                            }
                            else
                            {
                                ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                            }
                        }
                        else
                        {
                            ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                        }
                    }
                    else
                    {
                        ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                    }
```

```
                                }
                                else
                                {
                                    ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                                }
                            }
                            else
                            {
                                ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                            }
                        }
                        else
                        {
                            ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                        }
                    }
                    else
                    {
                        ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                    }
                }
                else
                {
                    ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                }
            }
            else
            {
                ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll
no. should be unique...');", true);
            }
        }
        else
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no.
should be unique...');", true);
        }

    }
    else
    {
```

```
                ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no. should be unique...');", true);
            }
        }
        else
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no. doesnt exist...');", true);
        }



    }

    //end of football grp event


    //start cricket.....

    protected void grpbtnbackpc1_Click(object sender, EventArgs e)
    {
        MultiView1.ActiveViewIndex = 0;
    }
    //checking if rollno exist for group registeristion
    protected void pc1_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@roll", pc1.Text);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc1.BorderColor = Color.Green;
        }
        else
        {
            pc1.BorderColor = Color.Red;

        }
    }
    protected void pc2_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc2.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc2.BorderColor = Color.Green;
        }
        else
        {
           pc2.BorderColor = Color.Red;

        }
    }
    protected void pc3_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc3.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc3.BorderColor = Color.Green;
        }
        else
        {
           pc3.BorderColor = Color.Red;
        }
    }
    protected void pc4_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc4.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc4.BorderColor = Color.Green;
```

```
        }
        else
        {
            pc4.BorderColor = Color.Red;
        }
    }
    protected void pc5_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc5.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc5.BorderColor = Color.Green;
        }
        else
        {
            pc5.BorderColor = Color.Red;
        }
    }
    protected void pc6_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc6.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc6.BorderColor = Color.Green;
        }
        else
        {
            pc6.BorderColor = Color.Red;
        }
    }
    protected void pc7_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc7.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc7.BorderColor = Color.Green;
        }
        else
        {
           pc7.BorderColor = Color.Red;
        }
      }
      protected void pc8_onTextchanged(object sender, EventArgs e)
      {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc8.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc8.BorderColor = Color.Green;
        }
        else
        {
           pc8.BorderColor = Color.Red;
        }
      }
      protected void pc9_onTextchanged(object sender, EventArgs e)
      {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc9.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc9.BorderColor = Color.Green;
        }
```

```
        else
        {
            pc9.BorderColor = Color.Red;
        }
    }
    protected void pc10_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc10.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc10.BorderColor = Color.Green;
        }
        else
        {
            pc10.BorderColor = Color.Red;
        }
    }
    protected void pc11_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc11.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc11.BorderColor = Color.Green;
        }
        else
        {
            pc11.BorderColor = Color.Red;
        }
    }
    protected void pc12_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
```

```
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc12.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc12.BorderColor = Color.Green;
        }
        else
        {
           pc12.BorderColor = Color.Red;
        }
    }
    protected void pc13_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc13.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc13.BorderColor = Color.Green;
        }
        else
        {
           pc13.BorderColor = Color.Red;
        }
    }
    protected void pc14_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll and courseid=@courseid", con);
        cmd.Parameters.AddWithValue("@courseid", Deptlist.SelectedValue.ToString());
        cmd.Parameters.AddWithValue("@roll", pc14.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
           pc14.BorderColor = Color.Green;
        }
        else
        {
```

```
            pc14.BorderColor = Color.Red;
        }
    }
    protected void pc15_onTextchanged(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from dummystudenttable where
studentid=@roll", con);
        cmd.Parameters.AddWithValue("@roll", pc15.Text);
        int i = (int)cmd.ExecuteScalar();
        con.Close();
        if (i > 0)
        {
            pc15.BorderColor = Color.Green;
        }
        else
        {
            pc15.BorderColor = Color.Red;
        }
    }
    //grp sport confirmation
    protected void grpbtnconfirmpc1_Click(object sender, EventArgs e)
    {
        if ((pc1.BorderColor == Color.Green) && (pc2.BorderColor == Color.Green) &&
(pc3.BorderColor == Color.Green) && (pc4.BorderColor == Color.Green) &&
(pc5.BorderColor == Color.Green) && (pc6.BorderColor == Color.Green) &&
(pc7.BorderColor == Color.Green) && (pc8.BorderColor == Color.Green) &&
(pc9.BorderColor == Color.Green) && (pc10.BorderColor == Color.Green) &&
(pc11.BorderColor == Color.Green) && (pc12.BorderColor == Color.Green) &&
(pc13.BorderColor == Color.Green) && (pc14.BorderColor == Color.Green) &&
(pc15.BorderColor == Color.Green))
        {
            //check p1 unique
            if ((pc1.Text != pc2.Text) && (pc1.Text != pc3.Text) && (pc1.Text != pc4.Text) &&
(pc1.Text != pc5.Text) && (pc1.Text != pc6.Text) && (pc1.Text != pc7.Text) && (pc1.Text !=
pc8.Text) && (pc1.Text != pc9.Text) && (pc1.Text != pc10.Text) && (pc1.Text != pc11.Text)
&& (pc1.Text != pc12.Text) && (pc1.Text != pc13.Text) && (pc1.Text != pc14.Text) &&
(pc1.Text != pc15.Text))
            {
                //check p2 unique
                if ((pc2.Text != pc3.Text) && (pc2.Text != pc4.Text) && (pc2.Text != pc5.Text)
&& (pc2.Text != pc6.Text) && (pc2.Text != pc7.Text) && (pc2.Text != pc8.Text) &&
(pc2.Text != pc9.Text) && (pc2.Text != pc10.Text) && (pc2.Text != pc11.Text) && (pc2.Text
!= pc12.Text) && (pc2.Text != pc13.Text) && (pc2.Text != pc14.Text) && (pc2.Text !=
pc15.Text))
                {
```

```
                //check p3 unique
                if ((pc3.Text != pc4.Text) && (pc3.Text != pc5.Text) && (pc3.Text != pc6.Text)
&& (pc3.Text != pc7.Text) && (pc3.Text != pc8.Text) && (pc3.Text != pc9.Text) &&
(pc3.Text != pc10.Text) && (pc3.Text != pc11.Text) && (pc3.Text != pc12.Text) && (pc3.Text
!= pc13.Text) && (pc3.Text != pc14.Text) && (pc3.Text != pc15.Text))
                {
                   //check p4 unique
                   if ((pc4.Text != pc5.Text) && (pc4.Text != pc6.Text) && (pc4.Text !=
pc7.Text) && (pc4.Text != pc8.Text) && (pc4.Text != pc9.Text) && (pc4.Text != pc10.Text)
&& (pc4.Text != pc11.Text) && (pc4.Text != pc12.Text) && (pc4.Text != pc13.Text) &&
(pc4.Text != pc14.Text) && (pc4.Text != pc15.Text))
                   {
                      //check p5 unique
                      if ((pc5.Text != pc6.Text) && (pc5.Text != pc7.Text) && (pc5.Text !=
pc8.Text) && (pc5.Text != pc9.Text) && (pc5.Text != pc10.Text) && (pc5.Text != pc11.Text)
&& (pc5.Text != pc12.Text) && (pc5.Text != pc13.Text) && (pc5.Text != pc14.Text) &&
(pc5.Text != pc15.Text))
                      {
                         //check p6 unique
                         if ((pc6.Text != pc7.Text) && (pc6.Text != pc8.Text) && (pc6.Text !=
pc9.Text) && (pc6.Text != pc10.Text) && (pc6.Text != pc11.Text) && (pc6.Text != pc12.Text)
&& (pc6.Text != pc13.Text) && (pc6.Text != pc14.Text) && (pc6.Text != pc15.Text))
                         {
                            //check p7 unique
                            if ((pc7.Text != pc8.Text) && (pc7.Text != pc9.Text) && (pc7.Text !=
pc10.Text) && (pc7.Text != pc11.Text) && (pc7.Text != pc12.Text) && (pc7.Text !=
pc13.Text) && (pc7.Text != pc14.Text) && (pc7.Text != pc15.Text))
                            {
                               //check p8 unique
                               if ((pc8.Text != pc9.Text) && (pc8.Text != pc10.Text) && (pc8.Text
!= pc11.Text) && (pc8.Text != pc12.Text) && (pc8.Text != pc13.Text) && (pc8.Text !=
pc14.Text) && (pc8.Text != pc15.Text))
                               {
                                  //check p9 unique
                                  if ((pc9.Text != pc10.Text) && (pc9.Text != pc11.Text) &&
(pc9.Text != pc12.Text) && (pc9.Text != pc13.Text) && (pc9.Text != pc14.Text) && (pc9.Text
!= pc15.Text))
                                  {
                                     //check p10 unique
                                     if ((pc10.Text != pc11.Text) && (pc10.Text != pc12.Text) &&
(pc10.Text != pc13.Text) && (pc10.Text != pc14.Text) && (pc10.Text != pc15.Text))
                                     {
                                        //check p11 unique
                                        if ((pc11.Text != pc12.Text) && (pc11.Text != pc13.Text)
&& (pc11.Text != pc14.Text) && (pc11.Text != pc15.Text))
                                        {
```

```
                                    //check p12 unique
                                    if ((pc12.Text != pc13.Text) && (pc12.Text != pc14.Text)
&& (pc12.Text != pc15.Text))
                                    {
                                       //check p13 unique
                                       if ((pc13.Text != pc14.Text) && (pc13.Text !=
pc15.Text))
                                       {
                                          //check p14 unique
                                          if ((pc14.Text != pc15.Text))
                                          {
                                             int token;
                                             //enter into db
                                             con.Open();
                                             SqlDataAdapter sd = new SqlDataAdapter("select
teamname from groupsportdetail", con);
                                             DataSet ds = new DataSet();
                                             sd.Fill(ds, "TNAME");
                                             List<string> tname = new List<string>();
                                             foreach (DataRow row in
ds.Tables["TNAME"].Rows)
                                             {
                                                tname.Add(row["teamname"].ToString());

                                             }
                                             con.Close();
                                             Boolean check = tname.Contains(teamname.Text);
                                             if (check == false)
                                             {
                                                con.Open();
                                                SqlCommand cmd2 = new SqlCommand("insert
into groupsportdetail (teamname,sportid,courseid,tournamentid) values
(@teamname,@sportid,@courseid,@tournamentid)", con);
                                                cmd2.Parameters.AddWithValue("@teamname",
teamname.Text);
                                                cmd2.Parameters.AddWithValue("@sportid",
toursportlist.SelectedValue.ToString());
                                                cmd2.Parameters.AddWithValue("@courseid",
Deptlist.SelectedValue.ToString());

cmd2.Parameters.AddWithValue("@tournamentid", tourlist.SelectedValue.ToString());
                                                cmd2.ExecuteNonQuery();
                                                con.Close();

                                                con.Open();
```

```
                                                SqlCommand cmd = new SqlCommand("select
token from groupsportdetail where teamname=@teamname ", con);
                                                cmd.Parameters.AddWithValue("@teamname",
teamname.Text);
                                                token = (int)cmd.ExecuteScalar();
                                                con.Close();
                                                string[] value = { pc1.Text, pc2.Text, pc3.Text,
pc4.Text, pc5.Text, pc6.Text, pc7.Text, pc8.Text, pc9.Text, pc10.Text, pc11.Text, pc12.Text,
pc13.Text, pc14.Text, pc15.Text };
                                                for (int i = 0; i < 15; i++)
                                                {
                                                   con.Open();
                                                   SqlCommand cmd1 = new
SqlCommand("insert into groupsporteventstudentlist (token,studentid) values
(@token,@studentid)", con);
                                                   cmd1.Parameters.AddWithValue("@token",
token);

cmd1.Parameters.AddWithValue("@studentid", value[i]);
                                                   cmd1.ExecuteNonQuery();
                                                   con.Close();
                                                }
                                                ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Succesfully registered');", true);
                                             }
                                             else
                                             {
                                                ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('team name already taken');", true);
                                             }
                                          }
                                          else
                                          {
                                             ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                          }
                                       }
                                       else
                                       {
                                          ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                       }
                                    }
                                    else
                                    {
```

```
                                        ScriptManager.RegisterStartupScript(this,
this.GetType(), "script", "alert('Roll no. should be unique...');", true);
                                    }
                                }
                                else
                                {
                                    ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                                }
                            }
                            else
                            {
                                ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                            }
                        }
                        else
                        {
                            ScriptManager.RegisterStartupScript(this, this.GetType(),
"script", "alert('Roll no. should be unique...');", true);
                        }
                    }
                    else
                    {
                        ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                    }
                }
                else
                {
                    ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                }
            }
            else
            {
                ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
            }
        }
        else
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
        }
```

```
                    }
                    else
                    {
                        ScriptManager.RegisterStartupScript(this, this.GetType(), "script",
"alert('Roll no. should be unique...');", true);
                    }
                }
                else
                {
                    ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll
no. should be unique...');", true);
                }

            }
            else
            {
                ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no.
should be unique...');", true);
            }
        }
        else
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no.
should be unique...');", true);
        }


    }
    else
    {
        ScriptManager.RegisterStartupScript(this, this.GetType(), "script", "alert('Roll no.
doesnt exist...');", true);
    }

  }
  //end of cricket grp event
    }
}
```

7. TESTING

Software testing is the crucial element of the software quality assurance and represents the ultimate review of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definitions and development phases, it was attempted to build software from an abstract concept to tangible information. The testing phase is a very important phase since it is in this phase; we make sure that the system will perform the task without any error. Testing is vital to the success of the system and is being done by classifying it in two ways- System Testing and Program Testing. Program Testing involves checking the syntax and logic of the program. This checking resulted in achieving error free programs.

No matter how carefully a programmer designs and plans application, the programs are sure to have a few bugs in them. Errors in the program immediately stop program execution and display an error message if the errors are syntax errors. After debugging one can identify the limitations of this project and hence corrections are made. During the system development, each source code was tested for its level of correctness. Each form was run a number of times in order to ensure that the details are entered correctly and works properly.

Testing methods

1. Unit Testing.
2. Black Box Testing.
3. White Box Testing
4. Integration Testing
5. Monkey Testing

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as "Module Testing". The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps each module is found to be working satisfactorily as regard to the expected output from the module.

This test can be considered as unit test. This has been carried out after the completion of one complete part. The word validation itself says about the nature of the test. Entire controls in the program have been tested in this manner. The limitations, nature and the boundaries are tested during the test. This test makes the work worthy to be developed further.

Black Box Testing

Black box testing methods focus on the requirements of the software that is black box testing enables the software engineer to drive sets of input conditions that will fully exercise all functional requirements for a program. It maintains the integrity of external information.

Black box testing attempts to find errors in the following categories.

- Incorrect or missing functions.
- Interface errors.
- Error in Data structures or external database access.
- Performance errors.

White Box Testing

White box testing of software is predicted on a close examination of procedural details. Logical paths through the software are tested by providing test cases that exercise specific sets of conditions and or loops. The status of the program may be examined at various points to determine if the expected or asserted status corresponds to the actual status. Using white box testing methods, the software engineer can derive test case.

- Guarantee that all independents paths within a module have been exercised at least once.
- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within the operational bounds.

Exercise internal data structures to ensure their validity

Integration Testing

Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems. We normally do Integration testing after "Unit testing". Once all the individual units are created and tested, we start combining those "Unit Tested" modules and start doing the integrated testing.

The main function or goal of this testing is to test the interfaces between the units/modules.

The individual modules are first tested in isolation. Once the modules are unit tested, they are integrated one by one, till all the modules are integrated, to check the combinational behavior, and validate whether the requirements are implemented correctly or not.

Here we should understand that Integration testing does not happen at the end of the cycle, rather it is conducted simultaneously with the development. So in most of the times, all the modules are not actually available to test and here is what the challenge comes to test something which does not exist!

Monkey Testing

Monkey testing is carried out by a tester assuming that if the monkey uses the application then how random input, values will be entered by the Monkey without any knowledge or understanding of the application. The objective of Monkey Testing is to check if an application or system gets crashed by providing random input values/data. Monkey Testing is performed randomly and no test cases are scripted and it is not necessary to

Monkey Testing is performed randomly and no test cases are scripted and it is not necessary to be aware of the full functionality of the system

8. IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project they are

- End user training
- End user Education
- Training on the application software
- System Design
- Parallel Run and to New System

End user Training:

The successful implementation of the new system will purely upon the involvement of the officers working in that department. The officers will be imparted the necessary training on the new technology.

End User Education:

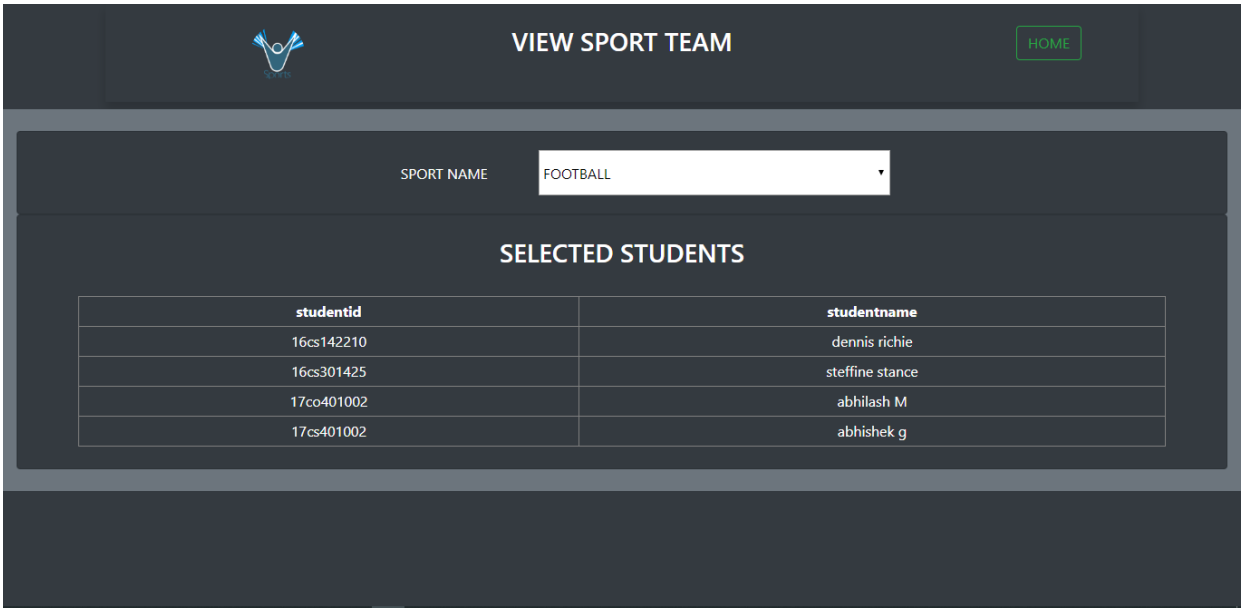
The education of the end user start after the implementation and testing is over. When the system is found to be more difficult to understand and complex, more effort is put to educate the end used to make them aware of the system, giving them lectures about the new system and providing them necessary documents and materials about how the system can do this.

Training of application software:

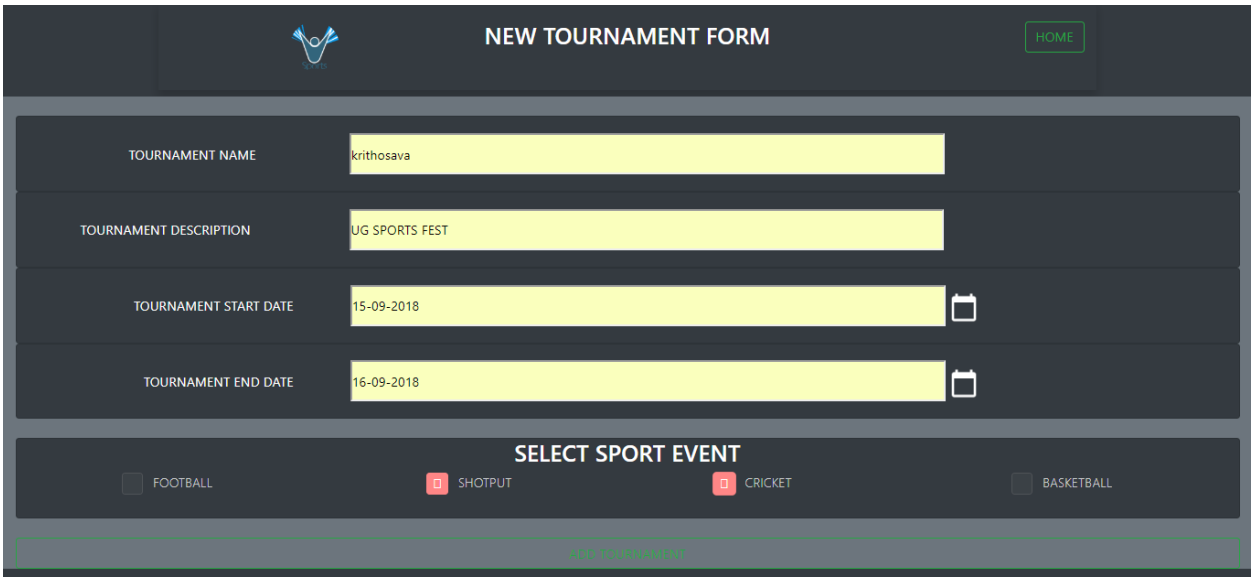
After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

9. SCREEN SHOT

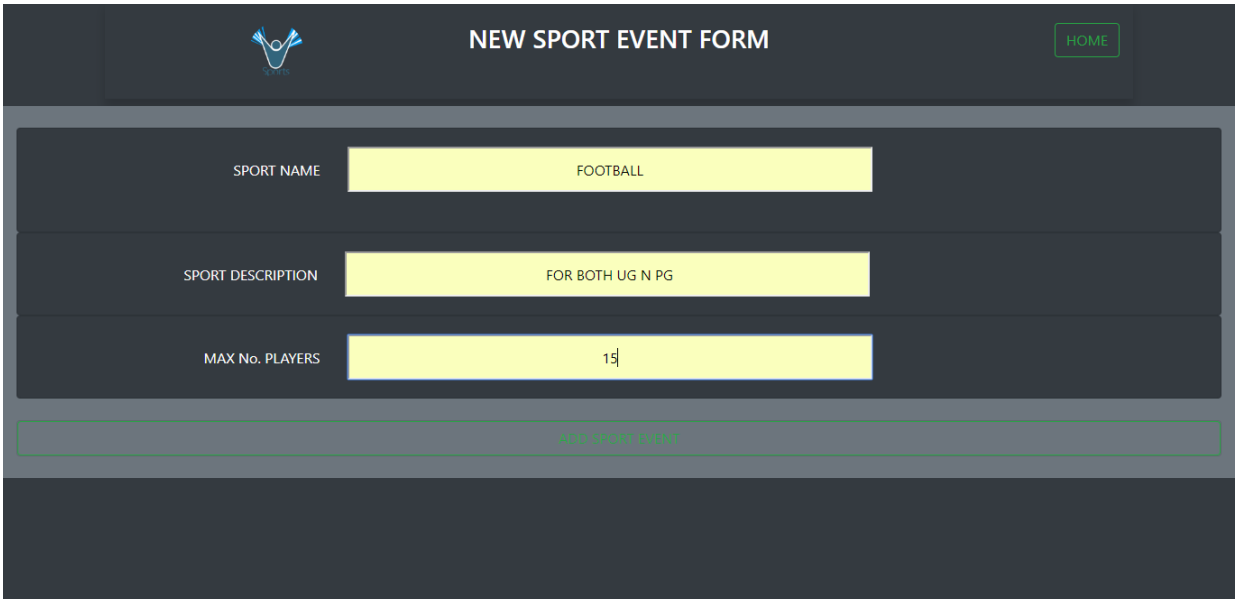
Sports team:



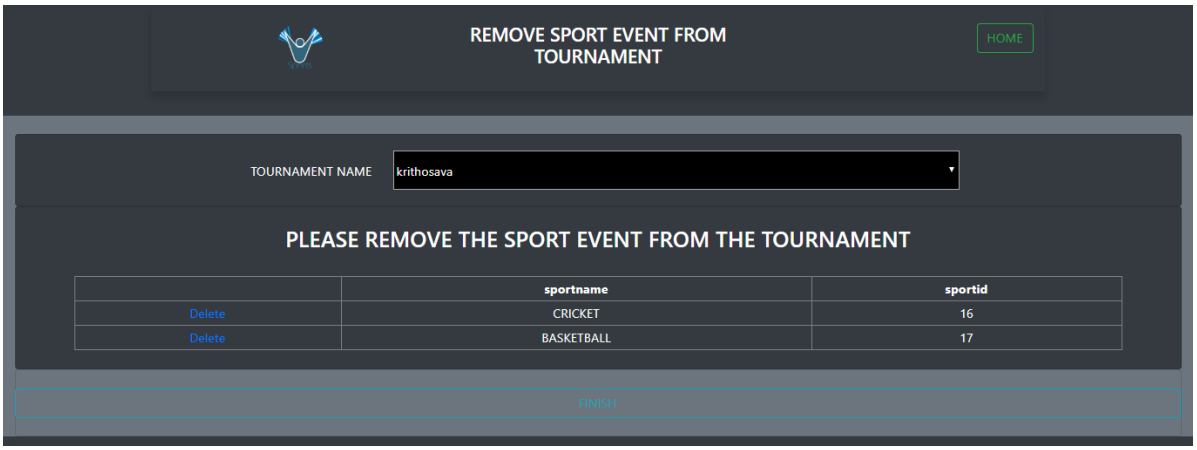
Adding new tournament:



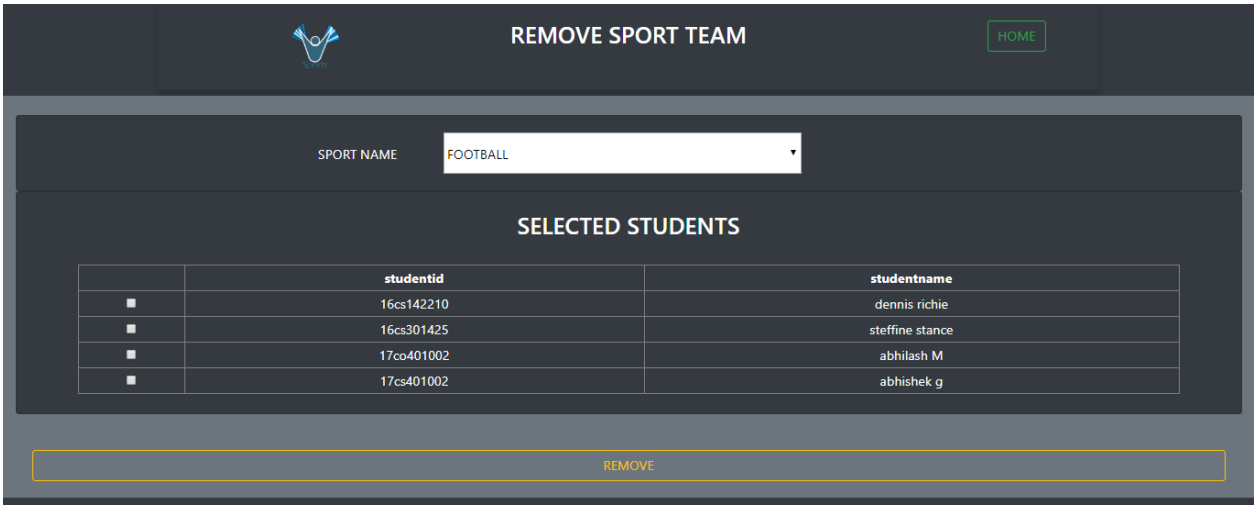
Adding a new sports:



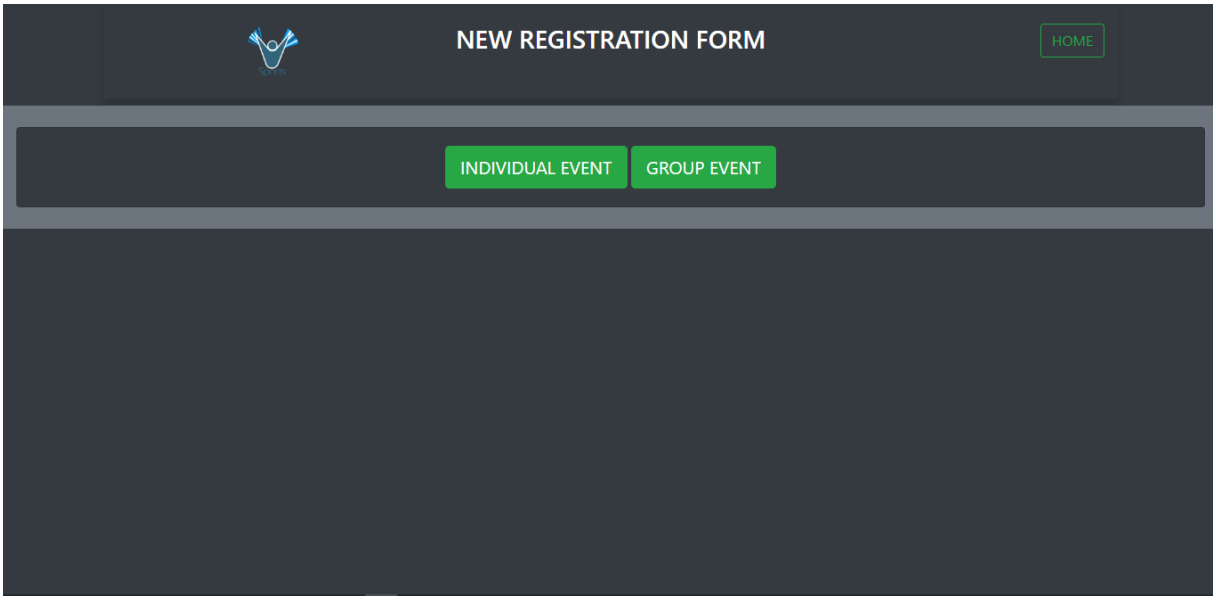
Removing a sports:



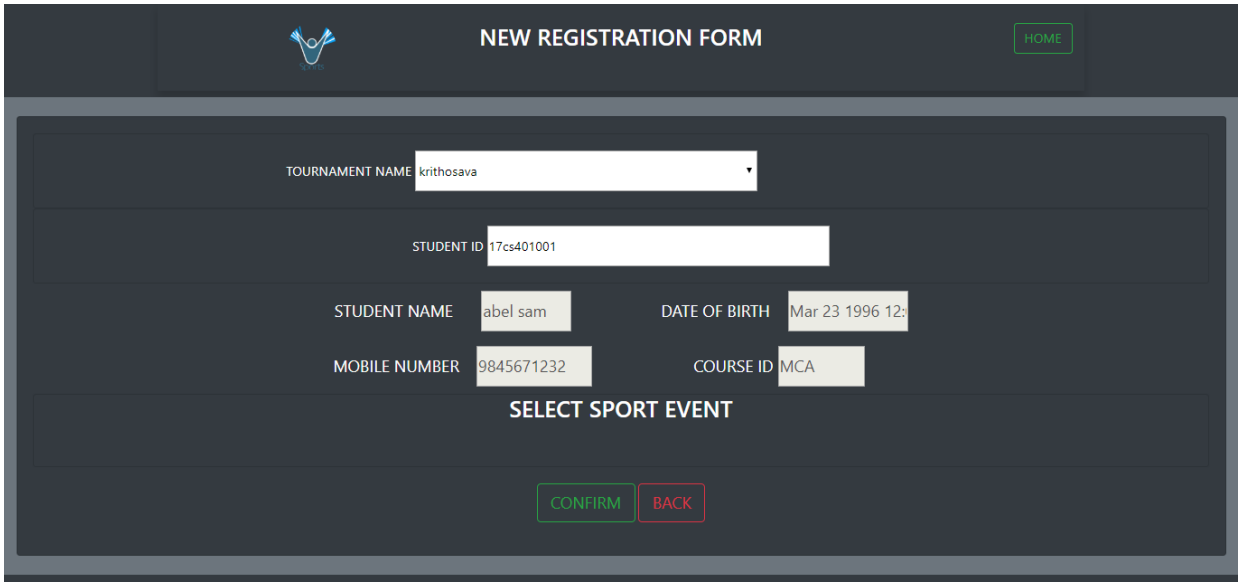
Removing Players:



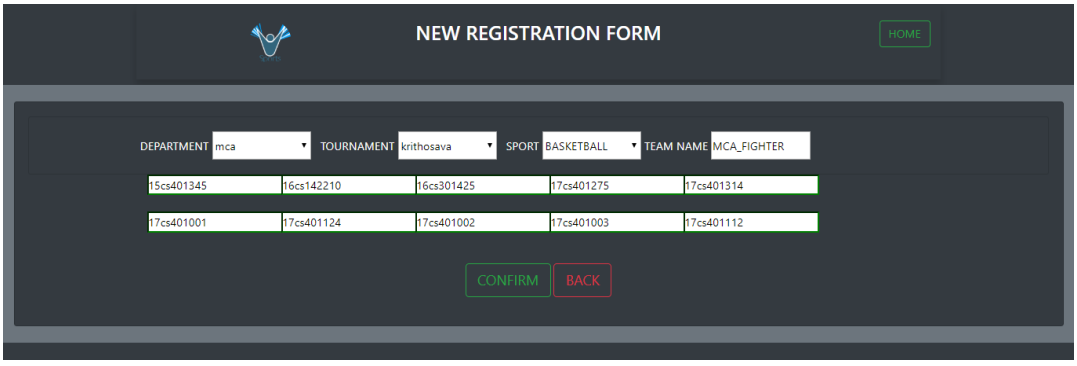
Index page for Registration:



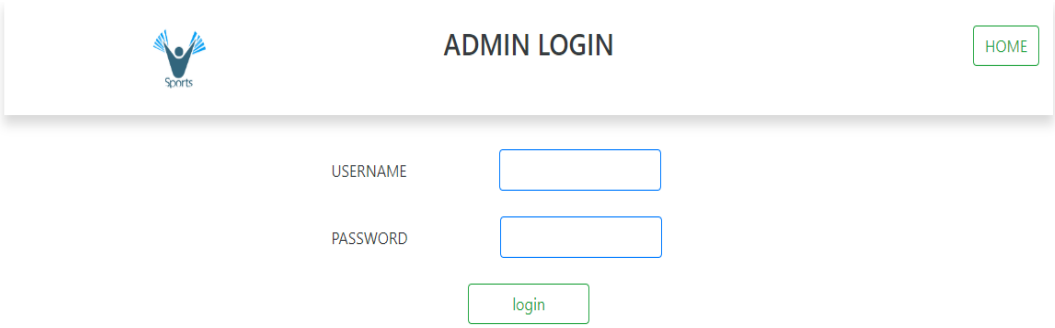
Registration form for intra-sports :



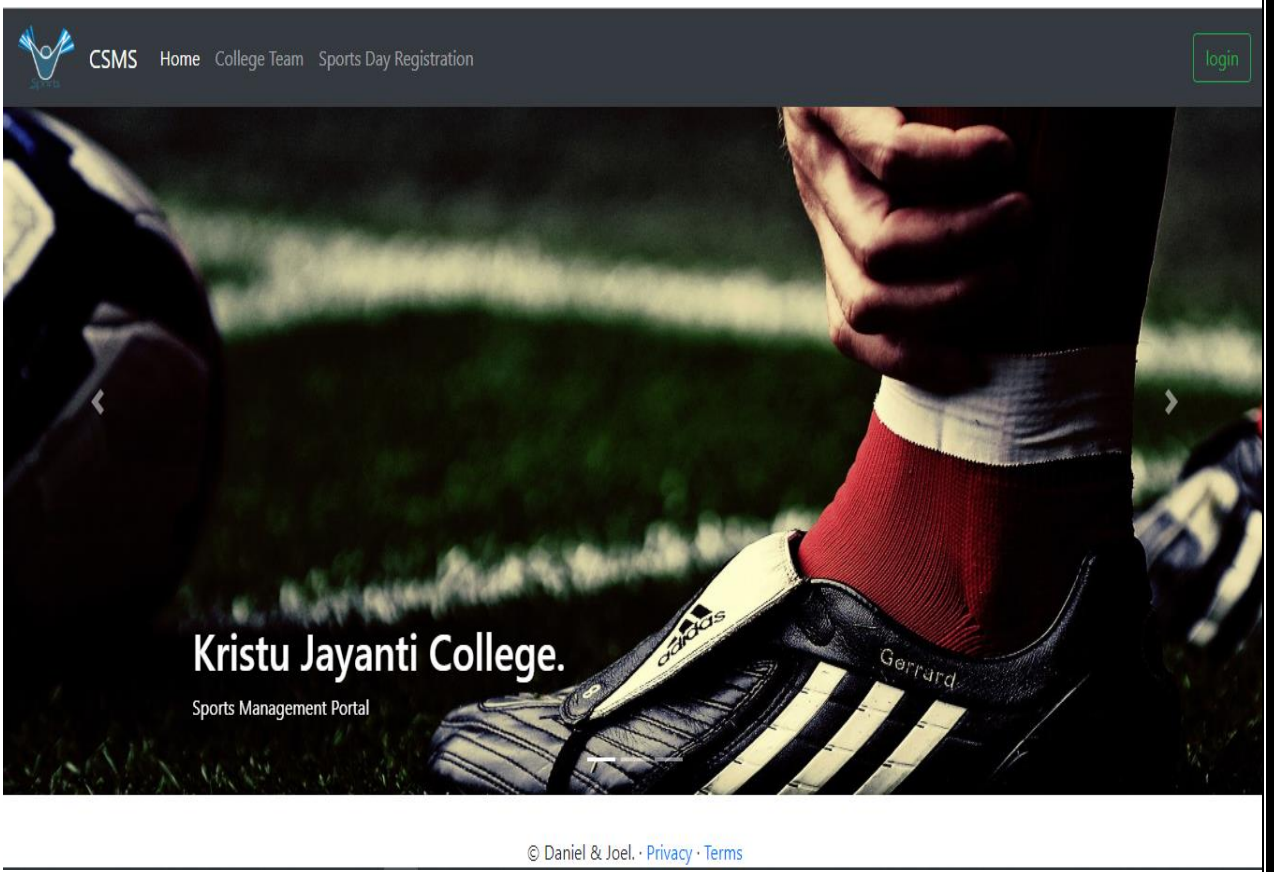
Registration form for group events:



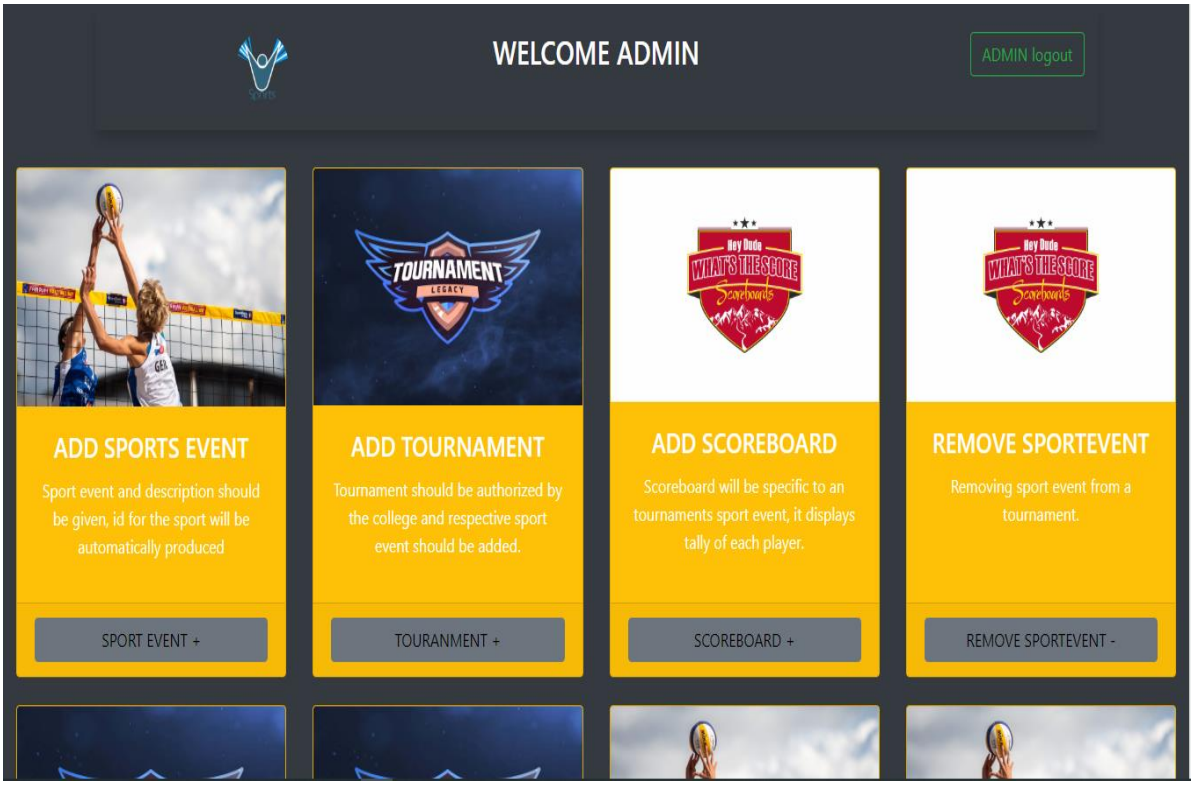
Admin Login:



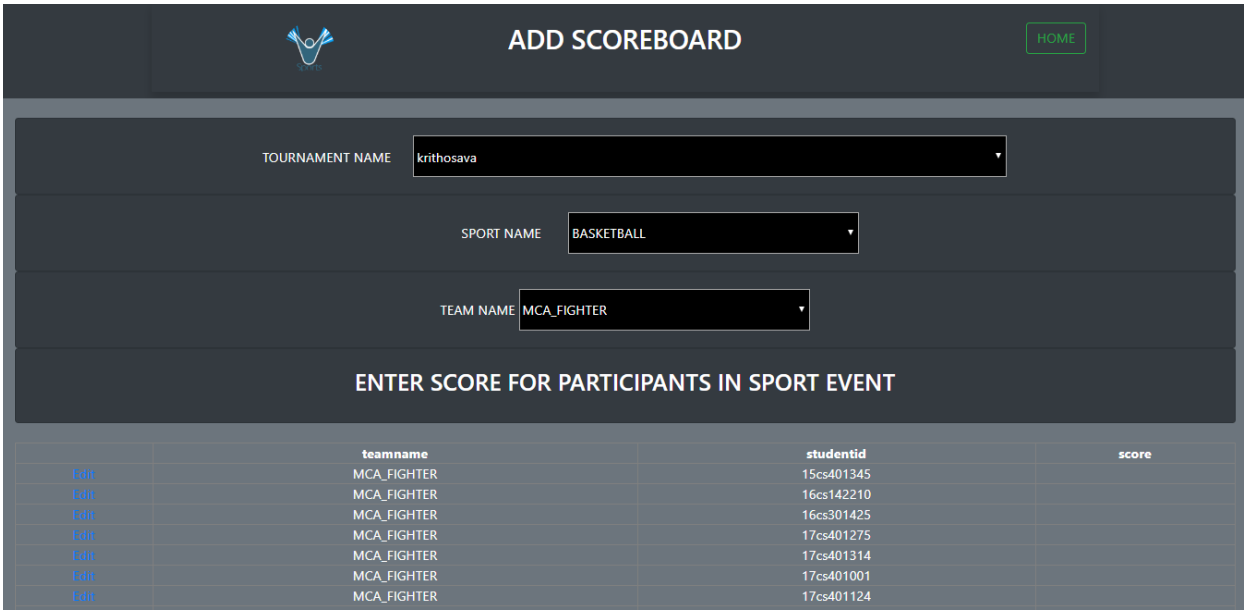
Index page:



Admin Home page:



Adding scoreboard:



10. CONCLUSION

Information Technology today is making incredible impact on our lives. Sports is a part of the curriculum in college. Hence it's time for the sports department and its activities to be automated and go online as it will be beneficiary for everyone who is associated with sports in college. The "**College Sports Management System**" project will assist the sports department of the college and would help the students in saving a lot of time in searching for games being conducted in the college. Our project provides students to get register from anywhere and anytime. It helps the administrative by streamlining the current intra-college sports event administrative practices. Specifically. It will allow the administrators to configure a set of Tournaments into the system and simply record the results of those contests so that administrators and the public can track the teams during the course of tournament.

.NET Framework makes the application robust, secure and reliable. This system provides better scalability and open to more enhancements. More level of abstraction can be implemented at front-end and back-end, thereby making the system easily adaptable to any changes in the environment.

The software which developed was implemented and tested with real data and was found to be error free. Also, it is found that the system works successfully. The user has to provide their personal detail to buy a package. All the necessary validations are carried out in this project, so that the company can make use of this software and necessary messages makes them conscious of the error they have made. Henceforth, valuable reports have been generated for this organization.

11. BIBOLOGRAPHY

Reference Book

- Roger S Pressman, **"Software Engineering",** 2000 Edition, Dreamtech Publications.
- Stephen Walther **"ASP.NET Kick Start"** 2002 Edition, Sams Publications
- Mathew MacDonald and Mario Szpuszta, **"ASP.NET 2.0 in C# 2005",** special Edition, Dreamtech Press**.**

Websites

- www.w3schools.com
- www.stackoverflow.com
- www.msdn.microsoft.com
- www.github.com
- Youtube.com
- Tutorialpoint.com