SPORTS MANAGEMENT SYSTEM

Submitted By

GOMATHI.K (17PCC005)

Under the guidance of

Mrs. D. SREEMATHI, M.Com (CA).,M.B.A., M.Phil., (Ph.D).,

In Fulfillment of the requirement for the award of the degree of

Master of Commerce with Computer Applications

DEPARTMENT OF COMMERCE

AVINASHILINGAM INSTITUTE FOR HOMESCIENCE AND

HIGHER EDUCATION FOR WOMEN

COIMBATORE-641043

APRIL 2019

CERTIFICATE

This is to certify that the project work entitled "SPORTS MANAGEMENT SYSTEM" Submitted to Department of Commerce, Avinashilingam Institute For Home Science and Higher Education For Women, Coimbatore, in partial fulfillment MASTER OF COMMERCE WITH COMPUTER APPLICATIONS, is the record of the original project work done by GOMATHI.K during the period of her study, under my supervision and guidance.

Signature of the Dean

D. Geetha

Signature of the Head of the Department

Submitted for the viva voice examination held on 30-04-2019

Signature of the Supervisor

Signature of the External Examiner

Avinashilingam
Institute for Home Science and Higher Education for Women
Deemed to be University Under category 'A' By MHRD, (Estd. u/s 3 of UGC Act 1956)
Re Accredited with 'A' Grade By NAAC, Recognised by UGC Under Section 12 B
Coimbatore - 641043, Tamil Nadu, India

Coimbatore

19-04-2019

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. K.GOMATHI (REGNO:17PCC005)** doing final year **M.Com(CA) ofAVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN, Coimbatore** has successfully completed her major project work entitled **"SPORTS MANAGEMENT SYSTEM"** during**December 2018 to April 2019.** She did her project well.

Assistant Director,

Physical Education.

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN · DEPARTMENT OF PHYSICAL EDUCATION · COIMBATORE TAMILNADU

Dr.P.NANDHINI M.P.Ed., Ph.D., PGDFM.,PGDYEd.,
Assistant Director of Physical Education
Avinashilingam Institute for Home Science
and Higher Education for Women
Coimbatore - 641 043

DECLARATION

I hereby declare that this project work entitled SPORTS MANAGEMENT SYSTEM submitted to Department of Commerce, Avinashilingam Institute for Home science and Higher Education for women, Coimbatore, in partial fulfillment of the requirement for the award of the degree of MASTER OF COMMERCE WITH COMPUTER APPLICATIONS is the bonafide record of original project work done by GOMATHI.K during the period of my study, under the supervision and guidance of Mrs.D.Sreemathi,M.Com(CA).,MBA., M.Phil.,(Ph.D).,

Place: Coimbatore

Date: 24-04-2019.

K.G.

Signature of the candidate

GOMATHI.K

ACKNOWLEDGEMENT

I would like to express my sincere thanks to **God Almighty**, for his constant love and grace that he has showed upon me, which kept me in good health, and sound mind without which my project would not have reached a successful end.

I would like to express my deep sense of reverential gratitude and sincere thanks to **Shri. Dr.D.Krishnakumar., Chancellor**, Avinashilingam Institute of Home Science and Higher Education for Women University, Coimbatore - 43, for the opportunity given to me for undertaking this study and for providing the needed facility during the course of my study.

I owe my great deal of gratitude to **Dr.(Mrs).PremavathyVijayan M.Sc., M.Ed., Dip.Spl.Edn., M.Phil., Ph.D.,Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my gratitude to **Dr.(Mrs).A.Kowsalya, Registrar**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I would express my boundless thanks to **Dr.(Mrs).U.Jerinabi,M.Com.,Dip.Ed.,M.Phil., Ph.D.,Dean School of Commerce and Management**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr.(Mrs).D.Geetha M.Com.,Dip.Ed., M.Phil.,Ph.D., Professor and Head**, Department of Commerce for providing all the facilities to complete the project.

I express my heart full gratitude to my mentor **Mrs.D.Sreemathi,M.Com[CA]., M.B.A.,M.Phil.,(Ph.D)., Assistant Professor,Department of Commerce,**for guidance and timely suggestions shown in aiding me to consummate this project work successfully.

Finally, yet importantly, I would like to thank **Dr.(Mrs).P.Nandhini, MPEd.,PGDFM.,PGDYEd.,Ph.D., andmy family members,friends** for their support, encouragement and prayers, which were instrumental in the successful completion of this project. I have great pleasure in expressing my deep sense of gratitude to all other teaching and non-teaching staff members who stood behind the screen for the completion of the project.I would

extend my hearty thanks to one and all that helped me directly or indirectly for the successful completion of my project.

LIST OF CONTENTS

	D. SCREENSHOT	

ABSTRACT

This project named “Sports Management System” has been created to maintain the student registration details and tournament details of the students in the University/College. Different students will be having interest in different sports. But sometimes the students may not be able to participate in such games since their names will not be in the participants list. This will happen only if the names are taken through the pen paper method. To solve these problems, the sports management system can play a very good role . The Project consists of various forms and developed with the help of the VB.Net as front end and MYSQL as back end.

The Sports Management System (SMS) objective is to provide which manages the activity of many sports at a time. It also manages the Sports student’s details and activity of students in college level. This application will also provide the information regarding a sports tournament and will provide all necessary details about the tournament. The admin will consume less amount of time when compared to manual paper work through the automated system. The system will provide the serving activity in quick and easy manner. The system also maintains information about the teams participating in the tournament and related to particular sports. The main aim of the project is to register the sports student details and the tournament details. The student’s information will be saved in the database depending on the particular sports. The software has a special feature of adding photos of particular students in student registration form. This software also provides to store the tournament date, venue, Result. Any information needed can be retrieved easily without any errors. The system is user friendly and error type. All the information will be saved safely in the database.

CHAPTER I

1. INTRODUCTION

Sports is one of the activities that most of the students like to participate whether it is at the school, college, district and state level. Different students will be having interest in different sports. But sometimes the students may not be able to participate in such games since their names will not be in the participants list. This will happen only if the names are taken through the pen paper method. To solve these problems, the sports management system can play a very good role. The students will be able to register their names in different sports of their choice at the college level or state level through this application.

The Sports Management System (SMS) objective is to provide which manages the activity of many sports at a time. It also manages the selection activity of students to college and to state level. 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. Sports management not only rewarding but it also helps the programmer to quickly organize the sports events and lists in short interval of time. 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.

1.1 OVERVIEW OF THE PROJECT

This software be produced on sports management system. There are 2 users' i.e., the admin and the employee. Admin provides username and password to the employee. He also has the right to add or modify the given username and password of the employee. Using this username and password, an employee can login to the system. The first procedure is the student registration. Here the admin enters all the details of a student including the sport they like to participate. All these information will be stored in the database. Next is the entry of the tournaments. Here depending on the sport, the tournament date and the venue is saved in the database for further confirmation. A list of players will be displayed and also the upcoming tournaments will be shown according to the particular sport. The system also

provides a special authority of adding photos to the system for a particular sport for sweet memories and for proof. The system also does the selection procedure to the college level and the state level competition. To the college level all the players are allowed to play whereas to the state level, the best players will be selected from the college level list which will be also saved in the database. Finally admin will take all the report from this application. So no paper work followed. All the works are done through systematically.

1.2 ABOUT THE INSTITUTE

Avinashilingam Institute for Home Science and Higher Education for women, Coimbatore was established as an aided college by the great patriot and Educationist **PadmaBhushan Dr.T.S.Avinashilingam** under the auspices of the Avinashilingam Education Trust in 1957 with 45 women student. It is now one of the largest institutions in the country to impart quality education for women at all levels. From time immemorial, the institutional activities have been focusing on providing education with **Academic Excellence, Vitality of culture and values and Social Relevance.**

The institute has introduced ‘sports ‘as part of curriculum which is unique of its kind in this region thereby promoting sports among women students in life. The sports and Tournament and Physical Education department of the Institute are fully equipped with state-of-art instruments facilities for indoor, outdoor, gymnastics, indigenous and traditional sports. The Institute provides opportunities and adequate facilities to pursue sports in equal important with regular studies before and after working hours in both the campus. The Institute also engages the special students in sports activities and in fitness classes with diversified provisions. Scholarships are offered for meritorious students to emulate them as great athletes and sports person.

The Vision and Mission of this Institute is

VISION

Self development and Empowerment of women through modern, Scientific and value based education to enable them to lead a purposeful life filled with moral and spiritual values.

MISSION

The Mission of the University is to provide quality education of global standards on a strong foundation of Indian values and traditions to women students based on current advances in science, technology and societal demands with emphasis on commitment to social progress, peace, harmony and national integration.

DEPARTMENT OF SPORTS

PROFILE

The department of Physical Education and sports was established in the year 1957 for which K.Vargese was appointed as physical Director. She was succeeded by Dr.T.Vimala Ko Sasi till the year 2013. Dr.P.Nandhini took over as Assistant Director of Physical Education from the year 2014 to till date.

VISION

"An Active and Winning women"

MISSION

To transform the delivery of sport and recreation by ensuring equitable access, development and excellence at all levels of participation and to harness the values that can create a better life for all women.

CORE VALUES

- Team Spirit
- Innovation
- Inclusion
- Integrity
- Responsibility
- Transparency
- Overcoming Adversity
- Tolerance and respect

It is easy to make Sports and physical fitness a regular part of the undergraduate experience at Avinashilingam Institute. We motivate our students to involve in some kind of sports-and nearly everyone joins in cheering on their friends and classmates.

Avinashilingam Varsity teams include basketball, Handball, volleyball, football, Badminton, Table Tennis, Hockey, Kabaddi, Kho-Kho, Chess, Yoga, Mallakhamb, Silambam, Taekwondo, fencing, Judo, cross-country and track and field.

SPORTS FACILITIES

Sports facilities are utilized by conducting national and state level tournaments/summer camps during vacation also for other institute/school students in and around the state.

1. OUTDOOR SPORTS

S.No	Sport/ Game	Number of courts/ fields Total Area in Square Meters			
		Main Campus		Satellite Meters	
		Number of court	Size (Sq.m)	Number of court	Size (Sq.m)
1	400 mts standard Track with 8 lanes	_	_	1	19,425
2	Basketball(Synthetic, concrete & Mud)	1	480	2	960
3	Ball Badminton	1	350	1	350
4	Kabaddi	2	400	1	200
5	Kho-Kho	2	896	1	448
6	Handball with fives Football	1	800	1	800
7	Throwball	1	266	1	266
8	Netball	1	480	-	-
9	Tennis	1	430	-	-
10	Volleyball	2	912	1	456
11	Football & cricket Field	-	-	1	5,000

2. INDOOR FACILITIES

S.No	Sport/ Game	Number of courts/ fields Total Area in Square Meters			
		Main Campus		Satellite Meters	
		Number of court	Size (Sq.m)	Number of court	Size (Sq.m)
1	Multi-Purpose Hall	_	_	1	1060
2	Kabaddi Mat	1	150	-	-
3	Table Tennis	1	15	2	30
4	Badminton	2	196	2	196
5	Power lifting	-	-	1 Set	16
6	Weight lifting	-	-	1 Set	16
7	Provisions for playing Chess, Carrom, Fencing, Takewondo, Gymnastics, Karate and Judo.				

SPORTS ACHIVEMENTS

The student have achieved many sports tournament in International, National, State, District level. The achievements list of students and also the students name list who are placed in sports quota are listed below.

International level

- N.Alamelu of III BPE won 8 medals in 3rd World Silambam Championship at Malaysia -2015.

- A.Elakkiya Thendral of II BSc Physical Education got 2nd place in Kukkiwon Cup Indian International Taekwondo Championship-2017 at New Delhi from 16/06/2017 to 18/06/2017.
- P.Saranya of III BPE won Silver Medal in the Foil Event and S.Sumithra of III BPE won Silver Medal in the Epee event in the I South Asian Fencing.
- M.Sudha of BPEd got 11th place International level "Mumbai Standard Charted Marathon" -2013 National level.

National Level

- C.Bhuvana of I BPED and P.Anusha , K.Agalya of II BSc Physical Education crowned gold medal in 6th rural Nationals 2016 in volleyball.
- National Handball Championships in 2017.

State Level

- **"1st Avinashilingam Ayya Memorial Trophy"-** State level volleyball tournament – Runners (2016).
- State Inter-Deemed University Over All Champions- Athletics (2014 &2015).

PLACEMENTS UNDER SPORTS QUOTA

- S.Menaka(Kabaddi Player) - Police Department, Govt. of TamilNadu.
- R.Rajeshwari (Kabaddi Player) - Police Department, Govt. of TamilNadu.

1.4 PROBLEM OF THE STATEMENT

The main problem of this project is planning, organization, time management on paper use of resource, and budget allocation of resource and space management. The effect is on management, organizations, workers and students.The impact of this problem is not proper completion of sports events at its specific time.

1.5 QUALITY ASSURANCE

Quality assurance managers play a crucial role in business by ensuring that products meet certain thresholds of acceptability. They plan, direct or coordinate quality assurance

programs and formulate quality control policies. They also work to improve an organization's efficiency and profitability by reducing waste.

CHAPTER II

SYSTEM CONFIGURATION

2.1 HARDWARE SPECIFICATION

Processor	:	Intel Dual core
Memory	:	1GB RAM
Hard Disk	:	80 GB HDD

2.2 SOFTWARE SPECIFICATION

Operating System	:	Windows XP
Front end	:	VB.NET
Back end	:	SQL Server

2.3 SOFTWARE DESCRIPTION

VISUAL BASIC .NET

Visual Basic .NET (**VB.NET**) is a multi-paradigm, object-oriented programming language, implemented on the .NET Framework. Microsoft launched VB.NET in 2002 as the successor to its original Visual Basic language. Although the ".NET" portion of the name was dropped in 2005, this article uses "Visual Basic [.NET]" to refer to all Visual Basic languages released since 2002, in order to distinguish between them and the classic Visual Basic. Along with Visual C#, it is one of the two main languages targeting the .NET framework.

VB.Net is a simple, modern, object-oriented computer programming language developed by Microsoft to combine the power of .NET Framework and the common language runtime with the productivity benefits that are the hallmark of Visual Basic. This

tutorial will teach you basic VB.Net programming and will also take you through various advanced concepts related to VB.Net programming language.

Microsoft's integrated development environment (IDE) for developing in Visual Basic .NET language is Visual Studio. Most Visual Studio editions are commercial; the only exceptions are Visual Studio Express and Visual Studio Community, which are freeware. In addition, the .NET Framework SDK includes a freeware command-line compiler called vbc.exe. Mono also includes a command-line VB.NET compiler.

VB.Net (10.0)

In April 2010, Microsoft released Visual Basic 2010. Microsoft had planned to use Dynamic Language Runtime (DLR) for that release but shifted to a co-evolution strategy between Visual Basic and sister language C# to bring both languages into closer parity with one another. Visual Basic's innate ability to interact dynamically with CLR and COM objects has been enhanced to work with dynamic languages built on the DLR such as IronPython and IronRuby. The Visual Basic compiler was improved to infer line continuation in a set of common contexts, in many cases removing the need for the " _" line continuation characters. Also, existing support of inline Functions was complemented with support for inline Subs as well as multi-line versions of both Sub and Function lambdas

Visual Basic .NET, is the latest version of visual basic, includes many new features. The Visual Basic supports interfaces but not implementation inheritance. Visual basic.net supports implementation inheritance, interfaces and overloading. In addition, Visual Basic .NET supports multithreading concept.

The following reasons make VB.Net a widely used professional language −Modern, general purpose.

- Object oriented
- Component oriented
- Easy to learn
- Structured language
- It produces efficient programs
- It can be compiled on a variety of computer platforms
- Part of .Net Framework

SYNTAX

VB.NET uses statements to specify actions. The most common statement is an expression statement, consisting of an expression to be evaluated, on a single line. As part of that evaluation, functions or subroutines may be called and variables may be assigned new values. To modify the normal sequential execution of statements, VB.NET provides several control-flow statements identified by reserved keywords. Structured programming is supported by several constructs including two conditional execution constructs (If… Then…Else…End If and Select Case… Case… End Select) and three iterative execution (loop) constructs (Do…Loop…For…To, and For Each). The For… To statement has separate initialization and testing sections, both of which must be present. The For Each statement steps through each value in a list.

STRONG PROGRAMMING FEATURES VB.NET

VB.Net has numerous strong programming features that make it endearing to multitude of programmers worldwide. Let us mention some of these features −Boolean Conditions

- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading

THE .NET FRAMEWORK

The .Net framework is a revolutionary platform that helps you to write the following types of applications −

- Windows applications
- Web applications
- Web services

The .Net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages: Visual Basic, C#, C++, Jscript, and COBOL, etc.

All these languages can access the framework as well as communicate with each other. The Net framework consists of an enormous library of codes used by the client languages like VB.Net. These languages use object-oriented methodology.

Following are some of the components of the .Net framework

- Common Language Runtime (CLR)
- The .Net Framework Class Library
- Common Language Specification
- Common Type System
- Metadata and Assemblies
- Windows Forms
- ASP.Net and ASP.Net AJAX
- ADO.Net
- Windows Workflow Foundation (WF)
- Windows Presentation Foundation
- Windows Communication Foundation (WCF)
- LINQ

WRITING VB.NET PROGRAMS ON LINUX OR MAC OS

Although the.NET Framework runs on the Windows operating system, there are some alternative versions that work on other operating systems. Mono is an open-source version of the .NET Framework which includes a Visual Basic compiler and runs on several operating systems, including various flavors of Linux and Mac OS. The most recent version is VB 2012.

The stated purpose of Mono is not only to be able to run Microsoft .NET applications cross-platform, but also to bring better development tools to Linux developers. Mono can be run on many operating systems including Android, BSD, iOS, Linux, OS X, Windows, Solaris and UNIX.

FEATURES OF VISUAL STUDIO .NET

ABOUT .NET FRAMEWORK

The .net framework is a new computing platform that simplifies application development in the highly distributed environment of the internet.

OBJECTIVES OF .NET FRAMEWORK

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
2. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications. To make communication on distributed environment to ensure that code be accessed by the .NET Framework can integrate with any other code.

COMPONENTS OF .NET FRAMEWORK

THE COMMON LANGUAGE RUNTIME (CLR)

The common language runtime is the foundation of the .NET Framework. It manages code at execution time, providing important services such as memory management, thread management, and remoting and also ensures more security and robustness. The concept of code management is a fundamental principle of the runtime. Code that target the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

THE .NET FRAME WORK CLASS LIBRARY

It is a comprehensive, object-oriented collection of reusable types used to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime to enables embeds managed components or Windows Forms controls in HTML documents.

FEATURES OF THE COMMON LANGUAGE RUNTIME

The common language runtime manages memory; thread execution, code execution, code safety verification, compilation, and other system services these are all run on CLR.

- **Security**
- **Robustness**
- **Productivity**
- **Performance**

SECURITY

The runtime enforces code access security. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich. With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin to perform file-access operations, registry-access operations, or other sensitive functions.

ROBUSTNESS

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The managed environment of the runtime eliminates many common software issues.

PRODUCTIVITY

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers.

PERFORMANCE

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS).

MYSQL

MySQL is an open source relational database management system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius 's daughter, and " SQL", the abbreviation for Structured Query Language .MySQL is free and open-source software under the terms of the gnu general public license , and is also available under a variety of proprietary licenses. mysql was owned and sponsored by the swedish company mysql ab, which was bought by sun microsystems (now oracle corporation). in 2010, when oracle acquired sun, widenius forked the open-source mysql project to create mariadb. mysql is a component of the lamp web application software stack (and others), which is an acronym for linux, apache, mysql, perl / php /python. mysql is used by many database-driven web applications, including drupal, joomla, phpbb, and wordpress . mysql is also used by many popular websites, including google (though not for searches), facebook, twitter,] flickr, and youtube .

OVERVIEW

MySQL is written in C and C++ . Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms , including AIX , BSDi , FreeBSD , HP-UX ,eComStation , i5/OS , IRIX , Linux , macOS ,Microsoft Windows , NetBSD , Novell NetWare ,OpenBSD , OpenSolaris, OS/2 Warp, QNX , Oracle Solaris ,

Symbian, SunOS , SCO OpenServer, SCOUnixWare , Sanos and Tru64 . A port of MySQL to OpenVMS also exists.

FEATURES

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures , using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Online Data Definition Language (DDL) when using the InnoDB Storage Engine.
- Information schema

A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine Transactions with save points when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.

DEPLOYMENT

MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.VB.Net MyAdmin . In the medium range, MySQL can be scaled by

deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are, however, limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server continually pushes binlog events to connected slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached , or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

MySQL Fabric is an integrated system for managing a collection of MySQL servers, and a framework on top of which high availability and database sharding is built. MySQL Fabric is open-source, and supports procedure execution in the presence of failure, providing an execution model usually called resilient execution. MySQL client libraries are extended so they are hiding the complexities of handling failover in the event of a server failure, as well as correctly dispatching transactions to the shards.

GRAPHICAL USER INTERFACES

A graphical user interface (GUI) is a type of interface that allows users to interact with electronic devices or programs through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. GUIs are easier to learn than command-line interfaces (CLIs), which require commands to be typed on the keyboard. It can perform various tasks such as creating, modifying or deleting databases, tables, fields or rows; executing SQL statements; or managing users and permissions. It can import data from CSV and SQL, and transform stored data into any format using a set of predefined functions, like displaying BLOB-data as images or download-links.

CHAPTER III

SYSTEM STUDY AND ANALYSIS

3.1 EXISTING SYSTEM

In the existing system the student details and the tournament details are maintained in the notebook. Drawback of this system is that Employee cannot able to filter the student name list in particular sports and also it occupies large amount of space in the system. The events oriented information can't reach the students properly. These tasks are time consuming. Most of the process is done through manual work so efficiency is less.

DISADVANTAGES

- Take too much of time.
- Updating and maintain information difficult task.
- Due to manual process less efficient and scheduling is very difficult task.
- Required information cannot be retrieved easily
- Waste of time for manual operations.
- Sometimes the operations done manually will prone to unsecured access.

3.2 PROPOSED SYSTEMS

Our proposed system is aimed to developing a windows and management process system to the institute to sports. In proposed system, the student information will be saved in the database and we can filter the students name in particular sports. If any tournament conducted in this institute, we can enter the tournament details through this application. The Report also generated through this application. The student name list can also view through their class-wise and along with their details. Our proposed system is user-friendly and it is used in efficient way.

ADVANTAGES

The objective of the project is to manage the resource to the full extend. There are several major reasons why institute undertakes software's

The proposed system has some advantages:

- Easy to use and fully automated process.
- Our scheme fully designed to reduce gap between participant and management.
- The admin can see the entire event details schedule, so the information is secure.
- The admin can effectively filter the player list in this application and the report can be easily generated.
- This system will provide the serving activity in quick and easy manner.

CHAPTER IV

SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4.1 FILE DESIGN

The file system is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into individual pieces, and giving each piece a name, the information is easily separated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a file. The structure and logic rules used to manage the groups of information and their names are called a "file system". There are many different kinds of file systems. Each one has different structure and logic, properties of speed, flexibility, security, size and more. Some file systems have been designed to be used for specific applications

4.2 INPUT DESIGN

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

- Student details can be Uploaded
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

4.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs.

In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

4.4 DATABASE DESIGN

The general theme behind a database is to handle information in an integrated manner. There is none of the artificiality that is normally embedded in separate files or applications. A database is collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user.

In a database environment, common data are available which several authorized users can use. The concept behind a database is an integrated collection of data and provides a centralized access to the data from the program. It makes possible to treat data as a separate resource.

While designing database, several objectives must be considered:

- Controlled redundancy
- Data Independence
- More information at low cost
- Accuracy and Integrity

- Recovery from failure
- Privacy and security
- Performance

STEPS FOR TABLE DESIGN

- State what kind of information we need to handle to get the desired output.
- Find out what information is needed for fields (i.e.) field type, size etc.
- Remove any data items, which is redundant.
- Table have one to one relationship needs a primary key field.
- Tables have one to many relationship needs to add a foreign key field to the table to match the primary key field table.

4.5 CODE DESIGN:

Code is an ordered collection of symbols designed to provide unique identification of an attribute. Codes can be used for various purposes. They can specify object's physical or performance characteristics and they can be used to give operational instructions. They also can show inter relationships and may sometimes use to achieve secrecy or confidentiality. Codes are designed for optimum human-oriented use and machine efficiency. Codes possess uniqueness, expandability, conciseness, uniform nets, simplicity, versatility, sort ability, meaningfulness and operability.

Sufficient effort and time is spent in the preliminary study of the problem to design an efficient code. Activate serve scripting is object oriented. The source code is designed so that it can do transaction efficiently. It is the code that dose all the updating, modifications, etc. for all object used in the project there exist an associated source code, which explains the work of that object. It also describes the flow of the project.

Source code is enhanced by structured coding techniques by good internal comments and features provided by the language.

The code design in this project is made modular. The modular behavior enables easy debugging and testing. Inserting comment statement wherever enhances the coding. This is done during the documentation process coding is done in such a way that errors can be trapped easily. Also modifications can easily be appended due to the codes modular behavior

CHAPTER V
SYSTEM DEVELOPMENT

5.1 MODULES

1. Authentication
2. Master
3. Games
4. Level
5. Report
6. Exit

5.2 MODULES DESCRIPTION

Authentication

Admin provides username and password in this module. The system will check the admin username and password. After verification admin login and made the further process. Using this username and password, an employee can login to the system.

Master

This module contains the Student Registration details form, Tournament details form, Tournament Registration details form, Tournament Result form, other college tournament participant details form.

1. Student Registration

The first procedure is the student registration. Here the admin/ Employee enter all the details of a student including the sports they like to participate. All these information will be stored in the database.

2. Tournament

Here depending on the sports, the tournament id, date and venue is saved in the database for further confirmation.

3. Tournament Registration

This module provides the tournament id, tournament name, student number, team name, student name, department and event name. The Admin will add these details in the system.

4. Tournament Result

This module provides the details automatically when clicking the tournament id, where it contains the tournament name, student number, team name, student name, event details. In this module the admin have to enter the overall percentage of the tournament and the result date will automatically displayed.

5. Other College

This module provides the tournament id, name, Student Name, Number, Department, Event and newly Token ID. The admin will add all these details in the system and the data is stored in database.

Games

In this module the players list will be displayed based on their games. Through the student registration the list of players in particular sports will be filter and displayed in this module.

Level

In this module the tournament result will be presented in graphical view. The system also does the selection procedure to the college level and state level competition. To the college level all the players are allowed to play. Whereas to the national level and state level, the best players will be selected from the college level which has been saved in the database.

Exit

This module allows the user to Exit the application. Further operations cannot be performed after user exits.

CHAPTER VI

SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

6.1.1 OBJECTIVES OF TESTING

Software testing is a critical element of software quality assurance that represents the ultimate review of specifications, design and coding. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data. It involves user training, system testing and successful running of the developed system.

The changes are made according to their needs. The testing phase involves the testing of the developed system using various kinds of data. While testing, errors are noted and corrections are made system testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The candidate system is subject to a variety of test: stress recovery, and security and usability tests.

6.1.2TEST PLAN

Testing is the process of executing a program with the intent of finding any errors. A good test of course has the high probability of finding a yet undiscovered error. A successful testing is the one that uncovers a yet undiscovered error.

A test is vital to the success of the system; system test makes a logical assumption that if all parts of the system are correct, then goal will be successfully achieved. The candidate system is subjected to a verity of tests online like responsiveness, its value, stress and

security. A series of tests are performed before the system is ready for user acceptance testing.

6.1.3 TESTING METHODS

The different types of testing are:-

- **Unit Testing**
- **Integration Testing**
- **Validation Testing**
- **Output Testing**
- **White box Testing**
- **Black box Testing**
- **User Acceptance 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 this step each module is found to be working satisfaction as regard to the expected output from the module.

INTEGRATION TESTING

Integration testing focuses on the design and construction of the software architecture. Data can be lost across an interface, one module can have adverse effect on another sub functions and show on. Thus integration testing is a systematic technique for constructing test to uncover errors associated with in the interface. In this project, all the modules are companied and then the entire program is tested as a whole.

VALIDATION TESTING

Validation testing is the requirement established as a part of software requirement analysis is validated against the software that has been constructed. This test provides the final assurance whether the software needs all functional, behavioral and performance requirements

Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactory.

OUTPUT TESTING

After performing the validation testing, the next step is the output testing of the proposed system, since no system could be useful if it does not produce required output in the specific format. Tested asking the users about the format required by them, the output is considered into two ways: one is on the screen and the other is printed format.

The output format on the screen is found to be correct as the format designed according to the user needs, for the hard copy also, the output comes as specified by the user. Hence output testing does not result in correction in the system.

WHITEBOX TESTING

White box Testing is done with the project which drive test cases that do the following

- Guarantee that all the independent paths with in modules have been exercise at least once.
- Exercise all logical decision on the true and false side.
- Execute all loops at the boundaries and within their operation bounds.
- Exercise internal data structures to ensure the validity

It is aimed at ensuring that the system works accurately and efficiently before live operation command.

BLACKBOX TESTING

Black box System methods focus on the functional requirement of the software. Using the black box testing method the following errors are identified and rectified in the package.

- Incorrect or Missing functions
- Interface Errors
- Errors in data Structures or external database access.

USER ACCEPTANCE TESTING

User acceptance testing of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep in touch with the prospective system user at time of developing and making changes wherever required.

6.2 SYSTEM IMPLEMENTATION

In this project, propose virtualizing Harvard architecture on top of the existing memory architecture of modern computers, including those without non-executable memory page support, so as to prevent the injection of malicious code entirely. Harvard architecture is simply one wherein code and data are stored separately. Data cannot be loaded as code and vice-versa. In essence, we create an environment where in any code injected by an attacker into a process' address space cannot even be addressed by the processor for execution.

In this way, we are attacking the code injection problem at its root by regarding the injected malicious code as data and making it unaddressable to the processor during an instruction fetch. Split memory architecture produces an address space where data cannot be fetched by the processor for execution. For an attacker attempting a code injection, this will prevent him from fetching and executing any injected code.

SYSTEM MAINTENANCE

The maintenance plan specifies the scheduled servicing tasks and intervals (preventive maintenance) and the unscheduled servicing tasks (adaptive or corrective maintenance). Tasks in the maintenance plan are allocated to the various maintenance agencies. A maintenance allocation chart is developed to tag the maintenance tasks to the appropriate maintenance agencies. These include: in-service or in-house work centers, approved contractors, affiliated maintenance or repair facilities, original equipment manufacturer, etc. The maintenance plan also establishes the requirements for the support resources.

Related activities such as resource planning, budgeting, performance monitoring, upgrades, longer term supportability, and sustenance also need to be managed. These activities are being planned, managed, and executed over a longer time horizon and they concern the wellbeing of the system over the entire life cycle. Proper maintenance of the system (including maintenance-free system designs) relies very much on the availability of support resources, such as support and test equipment, technical data and documentation, personnel, spares, and facilities. These have to be factored in during the acquisition agreement process.

CHAPTER VII
SCOPE OF FUTURE DEVELOPMENT

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in future when they required. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

CHAPTER VIII

CONCLUSION

The Sports Management System (SMS) objective is to provide which manages the activity of many sports at a time. It also manages the Sports student's details and activity of students in college level. This application will also provide the information regarding a sports tournament and will provide all necessary details about the tournament. The admin will consume less amount of time when compared to manual paper work through the automated system. So, the sports management system application was successfully created and stored all the **Authentication, Student details, Tournament participant details, Games details, Result details and Report details** into the database using this application. Several user friendly coding have used in this application. The application was tested very well and the errors were properly debugged. Testing also concluded that the performance of the system is satisfactory. All the necessary output is generated. This application shall prove to be a powerful package in satisfy all requirements of the user.

CHAPTER IX

BIBLIOGRAPHY

9.1 BOOKS REFERRED

- Alex Homer , **'Professional VB.NET 1.1'**, 2004 Edition, Wrox Publications
- Clayton crooks II **'Learning Visual Basic .Net Through Applications'**
- Roger S Pressman, **'Software Engineering'**, 2000 Edition, Dreamtech Publications
- Steven Holzner, **'Visual Basic.NET Black Book'**, 2003 Edition, Dreamtech Publications
- James Foxall '**Sams Teach Yourself Visual Basic ' ,**2015 1st Edition
- Henry F.Korth, '**Database System Concept'**, 1997 Third Edition, Megraw-Hill Publications
- John Smiley, '**Learn to program with VB.Net 2008 Express' ,** 2008 Edition

9.2 WEBSITES

- www.msdn.microsoft.com
- www.vbcity.com
- www.vbdotnetheaven.com
- www.codeproject.com
- www.dotnetjohn.com
- www.tutorialpoint.com

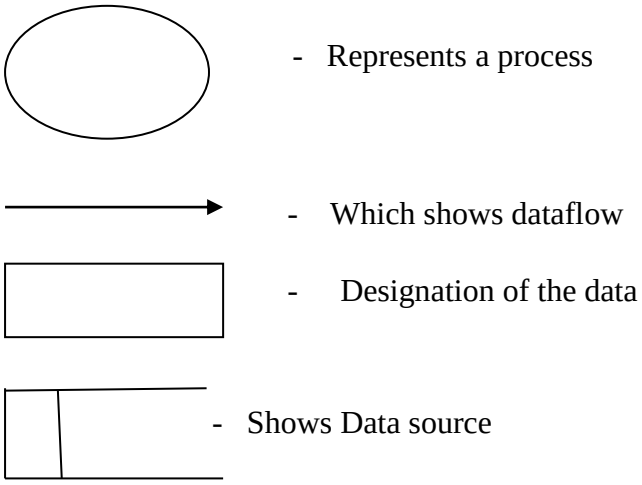
CHAPTER X

APPENDIX

A. DATA FLOW DIAGRAM

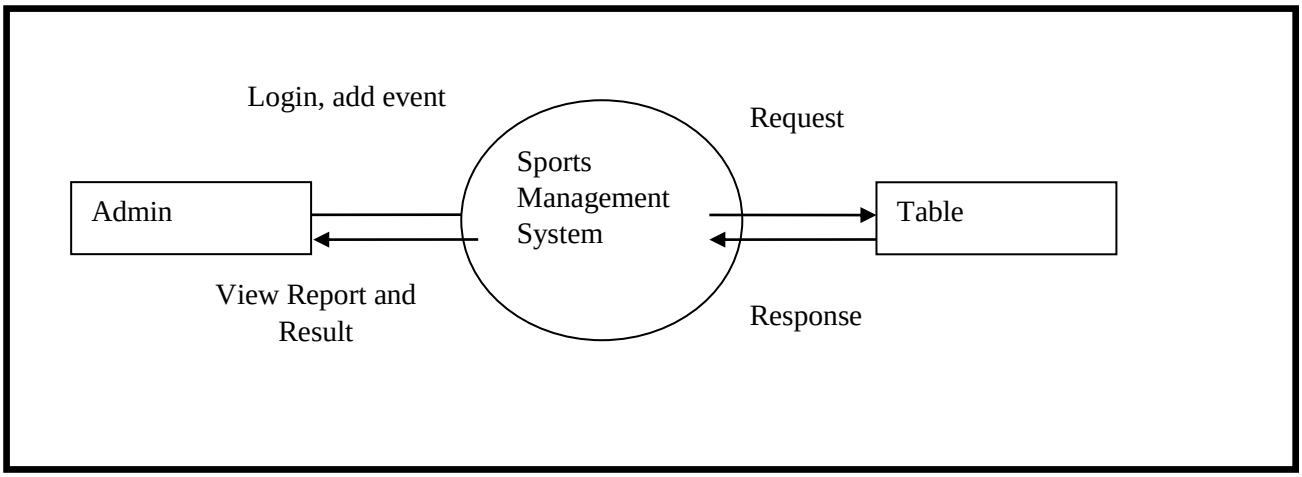
DFD depict hoe data interact with the system. DFD are extremely useful in modeling many aspects of a business function because they systematically subdivide a task into basic parts, helping the analyst understand the system that they trying to model data flow diagram models a system by using external entities from which data flow to a process which transmission the data and creates output data which goes to other processes on external entities of files. Data may also flow to process as inputs.

The symbols appearing in the DFD has been explained below:

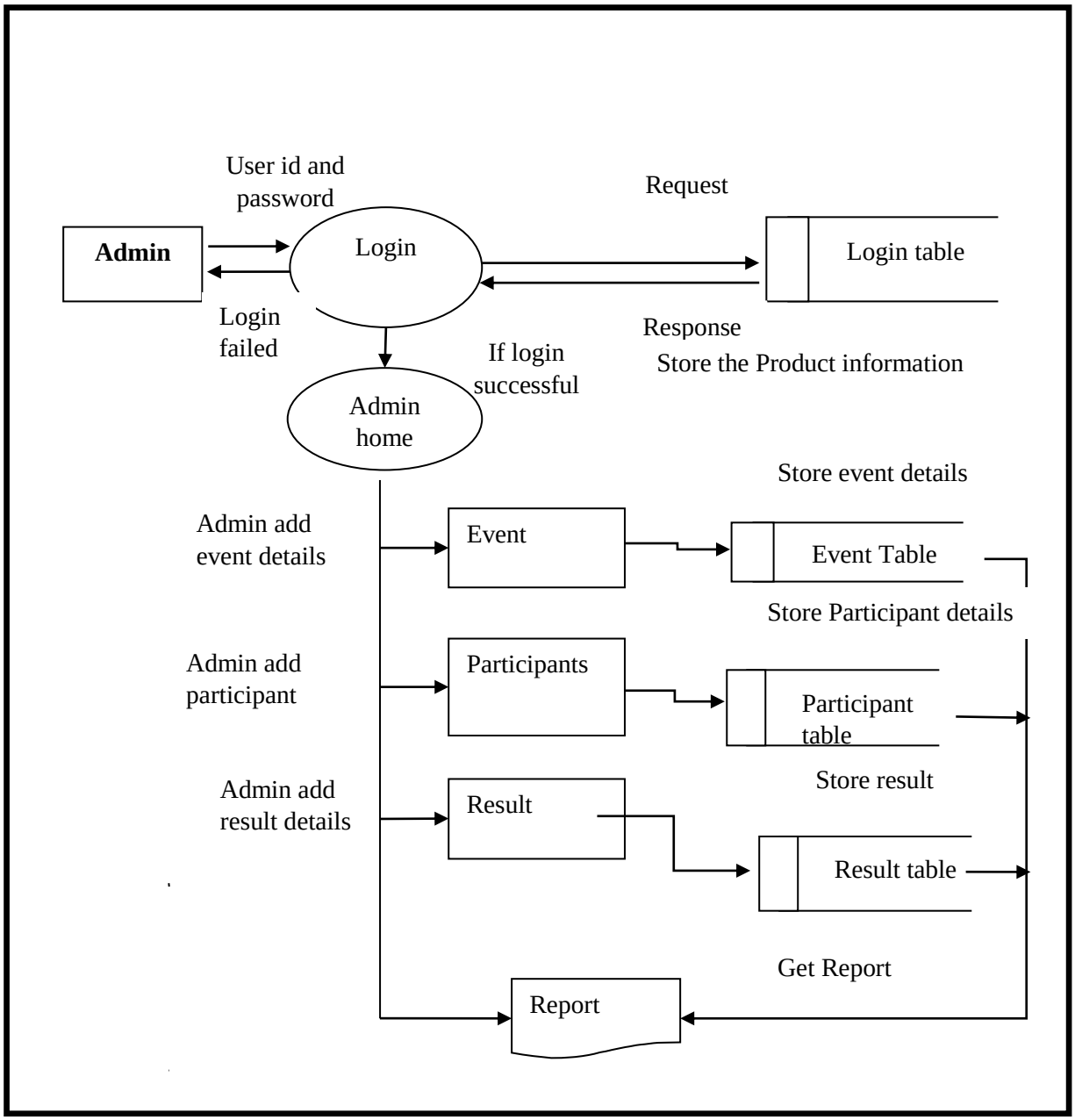


DATA FLOW DIAGRAM

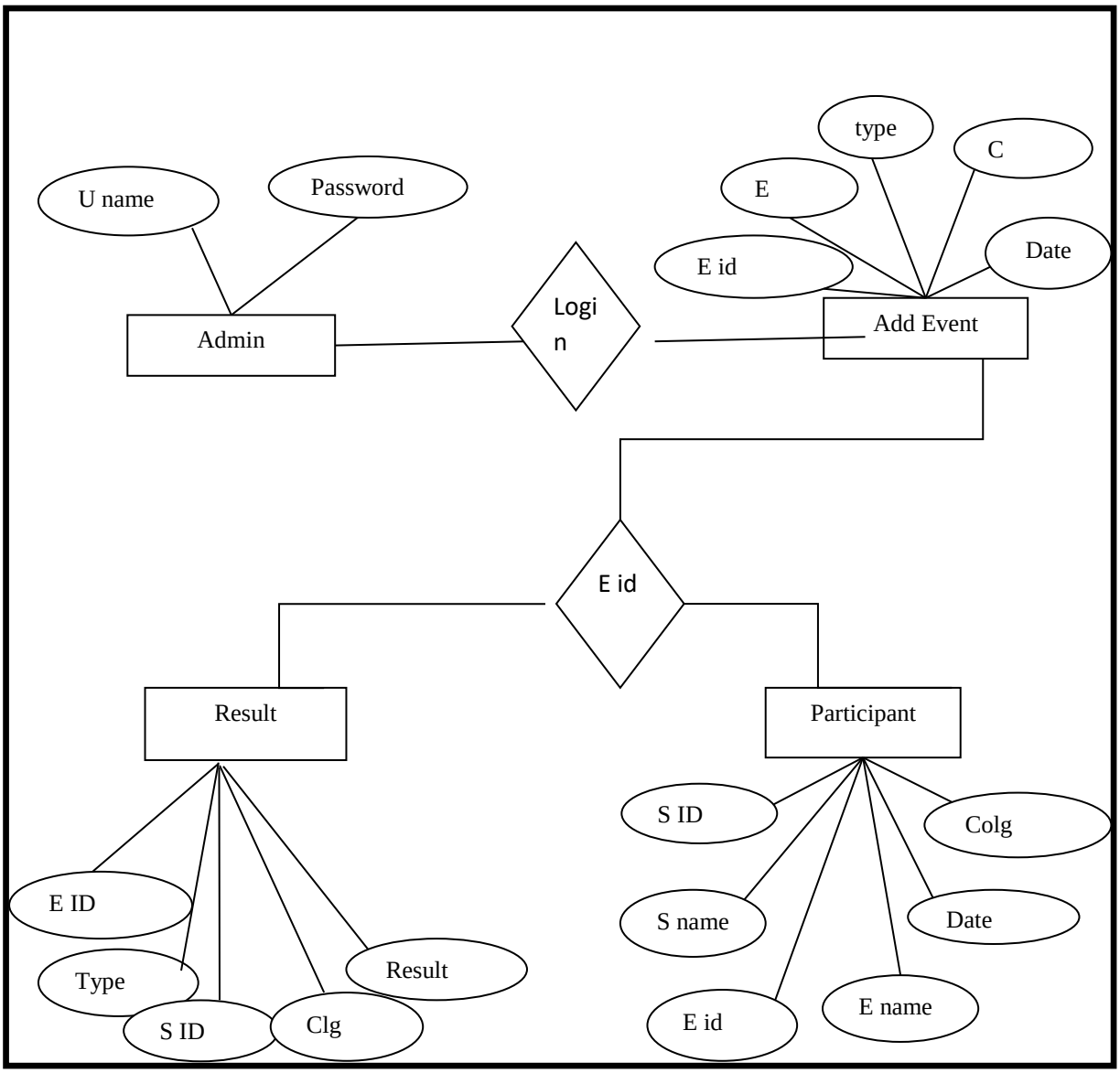
1. Level-0


Login, add event
Request
Sports
Management
System
Admin
Table
View Report and
Result
Response

2. Level-1


User id and password
Request
Admin
Login
Login table
Login failed
Response
If login successful
Store the Product information
Admin home
Store event details
Admin add event details
Event
Event Table
Store Participant details
Admin add participant
Participants
Participant table
Store result
Admin add result details
Result
Result table
Get Report
Report

B. ER DIAGRAM



C.TABLE STRUCTURE

1. Table Name: Student Registration

Primary key: sno

Fields	Data Type	Description	Constraints
Sno	Int	It contains Serial Number	Primary key
Sname	Varchar(50)	It contains Student Name	Not Null
Fname	Varchar(50)	It contains Father name	Not Null
Rno	Varchar(50)	It contains roll number	Not null
Class	Varchar(50)	It contains class	Not null
Batchnumber	Numeric(18, 0)	It contains batch number	Not Null
Department	Varchar(50)	It contains department	Not Null
Dob	Datetime	It contains date of birth	Not null
Age	Int	It contains age	Not null
Gender	Varchar(50)	It contains gender	Not Null
Mno	Numaeric(18,0)	It contains mobile number	Not Null
Ano	Numaeric(18,0)	It contains aadhar number	Not null
Event	Varchar(50)	It contains event name	Not null
Tournament	Varchar(50)	It contains tournament	Not Null
Community	Varchar(50)	It contains community	Not Null
Category	Varchar(50)	It contains category	Not null
Address	Varchar(50)	It contains address	Not null
Path	Varchar(MAX)	It contains image path	Not Null

2. Table Name: Tournament Registration

Primary key: tid

Fields	Data Type	Description	Constraints
Tid	Int	It contains tournament ID	Primary key
Tname	Varchar(50)	It contains tournament name	Not Null
Date	Datetime	It contains date	Not Null
Venue	Varchar(50)	It contains venue	Not null
Time	Varchar(50)	It contains time	Not null
Chiefguest	Varchar(50)	It contains chief guest	Not Null
Description	Varchar(50)	It contains description	Not Null

3. Table Name: Participant Tournament Registration

Foreign key: sno

Fields	Data Type	Description	Constraints
Tid	Int	It contains tournament id	Foreign key
Tname	Varchar(50)	It contains tournament name	Not Null
Sno	Int	It contains serial number	Not Null
Team	Varchar(50)	It contains team name	Not null
Sname	Varchar(50)	It contains student name	Not null
Department	Varchar(50)	It contains department	Not Null
Event	Varchar(50)	It contains event name	Not Null

4. Table Name: Tournament Result

Fields	Data Type	Description	Constraints
Tid	Int	It contains tournament id	Not Null
Tname	Varchar(50)	It contains tournament name	Not Null
Sno	Int	It contains serial number	Not Null
Team	Varchar(50)	It contains team name	Not null
Sname	Varchar(50)	It contains student name	Not null
Event	Varchar(50)	It contains event name	Not Null
Result1	Varchar(50)	It contains result	Not Null
Date	Datetime	It contains date	Not null

5. Table name: Other College Participant Registration

Primary key: tokenid

Fields	Data Type	Description	Constraints
Tid	Int	It contains tournament id	Primary key
Tname	Varchar(50)	It contains tournament name	Not Null
Cname	Varchar(50)	It contains college name	Not Null
Team	Varchar(50)	It contains team name	Not null
Sname	Varchar(50)	It contains serial number	Not null
Department	Varchar(50)	It contains department	Not Null
Event	Varchar(50)	It contains event name	Not Null
Tokened	Int	It contains token id	Not null

D. SAMPLECODINGS

1. LOGIN PAGE

```
Public Class Login
    Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        End
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "Admin" And TextBox2.Text = "Admin" Then
AdminHome.Show()
Me.Hide()
        Else
MsgBox("Invalid Username and Password!")
        End If
    End Sub
```

2. STUDENT REGISTRATION

```
Imports System.Data.SqlClient
Imports System.Data.Sql
Public Class StudentRegistration
    Dim ss As Integer
    Dim cmd As New SqlCommand
    Private Sub autonum()
con.Close()
con.Open()
        Dim cmd As New SqlCommand("select max(sno) from student", con)
        Dim dr As SqlDataReader
dr = cmd.ExecuteReader
        If (dr.Read) Then
            If IsDBNull(dr.Item(0)) Then
ss = 0
            Else
ss = dr.Item(0)
ss = ss + 0
            End If
        End If
con.Close()
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
cleartxt()
autonum()
        TextBox1.Text = ss + 1
    End Sub
    Private Sub cleartxt()
        TextBox1.Text = ""
        TextBox2.Text = ""
```

```
        TextBox3.Text = ""
        TextBox4.Text = ""
        ComboBox1.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
        TextBox9.Text = ""
        TextBox10.Text = ""
        TextBox11.Text = ""
        TextBox12.Text = ""
        ComboBox2.Text = ""
        ComboBox3.Text = ""
        ComboBox4.Text = ""
        ComboBox5.Text = ""
        ComboBox6.Text = ""
        ComboBox7.Text = ""
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        If (TextBox1.Text = "" And TextBox2.Text = "" And TextBox3.Text = "" And
TextBox4.Text = "" And ComboBox1.Text = "" And TextBox6.Text = "" And
TextBox7.Text = "" And TextBox8.Text = "" And TextBox9.Text = "" And
ComboBox2.Text = "" And ComboBox3.Text = "" And ComboBox4.Text = "" And
ComboBox5.Text = "" And ComboBox6.Text = "" And ComboBox7.Text = "") Then
MessageBox.Show("Please enter all values")
        Else
' Try
con.Close()
con.Open()
            cmd = New SqlCommand("insert into student values('" + TextBox1.Text + "','" +
TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + ComboBox1.Text +
"','" + TextBox6.Text + "','" + ComboBox2.Text + "','" + TextBox7.Text + "','" +
TextBox8.Text + "','" + ComboBox3.Text + "','" + TextBox9.Text + "','" + TextBox10.Text +
"','" + ComboBox4.Text + "','" + ComboBox5.Text + "','" + ComboBox6.Text + "','" +
ComboBox7.Text + "','" + TextBox11.Text + "','" + TextBox12.Text + "')", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Student Details Entered successfully")
grid()
cleartxt()
autonum()
            TextBox1.Text = ss + 1
con.Close()
' Catch ex As Exception
'MessageBox.Show("Enter Valid Input!")
' End Try
        End If
    End Sub
    Private Sub grid()
con.Close()
```

```
con.Open()
        Dim c As New SqlCommand("select * from student", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
con.Close()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim i As Integer
        i = InputBox("Enter Serial Number")
        Try
con.Close()
con.Open()
            Dim da As New SqlCommand("select * from student where sno='" + i.ToString() +
"'", con)
            Dim ds As SqlDataReader
ds = da.ExecuteReader
            While (ds.Read())
                TextBox1.Text = ds.Item(0)
                TextBox2.Text = ds.Item(1)
                TextBox3.Text = ds.Item(2)
                TextBox4.Text = ds.Item(3)
                ComboBox1.Text = ds.Item(4)
                TextBox6.Text = ds.Item(5)
                ComboBox2.Text = ds.Item(6)
                TextBox7.Text = ds.Item(7)
                TextBox8.Text = ds.Item(8)
                ComboBox3.Text = ds.Item(9)
                TextBox9.Text = ds.Item(10)
                TextBox10.Text = ds.Item(11)
                ComboBox4.Text = ds.Item(12)
                ComboBox5.Text = ds.Item(13)
                ComboBox6.Text = ds.Item(14)
                ComboBox7.Text = ds.Item(15)
                TextBox11.Text = ds.Item(16)
                TextBox12.Text = ds.Item(17)
                PictureBox1.ImageLocation = ds.Item(17)
            End While
grid()
con.Close()
ds.Close()
        Catch ex As Exception
MessageBox.Show("Invalid Serial Number!")
        End Try
    End Sub
```

```
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Try
con.Close()
con.Open()
            cmd = New SqlCommand("update student set sno='" + TextBox1.Text + "',sname='"
+ TextBox2.Text + "',fname='" + TextBox3.Text + "',rno='" + TextBox4.Text + "',class='" +
ComboBox1.Text + "',batchnumber='" + TextBox6.Text + "',department='" +
ComboBox2.Text + "',dob='" + TextBox7.Text + "',age='" + TextBox8.Text + "',gender='" +
ComboBox3.Text + "',mno='" + TextBox9.Text + "',ano='" + TextBox10.Text + "',event='" +
ComboBox4.Text + "',tournament='" + ComboBox5.Text + "',community='" +
ComboBox6.Text + "',category='" + ComboBox7.Text + "',address='" + TextBox11.Text +
"',path='" + TextBox12.Text + "' where sno='" + TextBox1.Text + "'", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Student details Updated")
grid()
cleartxt()
con.Close()
        Catch ex As Exception
MessageBox.Show("Enter Valid Information for Updation!")
        End Try
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        If TextBox1.Text = "" Then
MessageBox.Show("Enter the Serial Number")
        Else
            Try
con.Close()
con.Open()
                Dim cmd As New SqlCommand("delete from student where sno='" +
TextBox1.Text + "'", con)
cmd.ExecuteNonQuery()
MsgBox("Student details Deleted Successfully")
grid()
cleartxt()
con.Close()
            Catch ex As Exception
MessageBox.Show("Enter Valid Information for Deletion!")
            End Try
        End If
    End Sub

    Private Sub StudentRegistration_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
grid()
    End Sub
```

```
    Private Sub TournamentsToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
Tournament.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Football(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Cricket(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Hockey(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Volleyball(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Badminton(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Basketball(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub PlayersToolStripMenuItem6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Kabaadi(Players.DataGridView1)
Players.Show()
    End Sub

    Private Sub GameDevelopmemtLevelToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
College.Show()
```

```
    End Sub

    Private Sub LogoutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Login.Show()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        End
    End Sub

    Private Sub StudentRegistrationToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Me.Show()
    End Sub

    Private Sub TournamentRegistrationToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
TournamentRegistration.Show()
    End Sub

    Private Sub FootballToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub TournamentResultToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
Result.Show()
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
OpenFileDialog1.ShowDialog()
        PictureBox1.ImageLocation = OpenFileDialog1.FileName
        TextBox12.Text = OpenFileDialog1.FileName
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
cleartxt()
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
Me.Close()
    End Sub
```

```
    Private Sub DateTimePicker1_ValueChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles DateTimePicker1.ValueChanged
        Dim dob, d As Date
dob = DateTimePicker1.Value
        TextBox7.Text = dob
        d = Date.UtcNow
        Dim ts As TimeSpan
ts = d - dob
        Dim age As Integer
age = (ts.Days / 365) + 1
        TextBox8.Text = age
    End Sub
End class
```

3.TOURNAMENT DETAILS

```
Imports System.Data.SqlClient
Imports System.Data.Sql
Public Class Tournament
    Dim ss As Integer
    Dim cmd As New SqlCommand
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
Me.Close()
    End Sub
    Private Sub cleartxt()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
    End Sub
    Private Sub autonum()
con.Close()
con.Open()
        Dim cmd As New SqlCommand("select max(tid) from tournament", con)
        Dim dr As SqlDataReader
dr = cmd.ExecuteReader
        If (dr.Read) Then
            If IsDBNull(dr.Item(0)) Then
ss = 0
            Else
ss = dr.Item(0)
ss = ss + 0
            End If
        End If
con.Close()
    End Sub
```

```
   Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
cleartxt()
autonum()
       TextBox1.Text = ss + 1
   End Sub
   Private Sub grid()
con.Close()
con.Open()
       Dim c As New SqlCommand("select * from tournament", con)
       Dim da As New SqlDataAdapter(c)
       Dim ds As New DataSet
da.Fill(ds)
       DataGridView1.DataSource = ds.Tables(0)
con.Close()
   End Sub
   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
       If (TextBox1.Text = "" And TextBox2.Text = "" And TextBox3.Text = "" And
TextBox4.Text = "" And TextBox5.Text = "" And TextBox6.Text = "" And TextBox7.Text =
"") Then
MessageBox.Show("Please enter all values")
       Else
           Try
con.Close()
con.Open()
cmd = New SqlCommand("insert into tournament values('" + TextBox1.Text + "','" +
TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','"
+ TextBox6.Text + "','" + TextBox7.Text + "')", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Tournament Details Entered successfully")
grid()
cleartxt()
autonum()
               TextBox1.Text = ss + 1
con.Close()
           Catch ex As Exception
MessageBox.Show("Enter Valid Input!")
           End Try
       End If
   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
       Dim i As Integer
       i = InputBox("Enter Tournament ID")
       Try
con.Close()
con.Open()
```

```
        Dim da As New SqlCommand("select * from tournament where tid='" + i.ToString()
+ "'", con)
        Dim ds As SqlDataReader
ds = da.ExecuteReader
        While (ds.Read())
            TextBox1.Text = ds.Item(0)
            TextBox2.Text = ds.Item(1)
            TextBox3.Text = ds.Item(2)
            TextBox4.Text = ds.Item(3)
            TextBox5.Text = ds.Item(4)
            TextBox6.Text = ds.Item(5)
            TextBox7.Text = ds.Item(6)
        End While
grid()
con.Close()
ds.Close()
    Catch ex As Exception
MessageBox.Show("Enter Valid Information for Searching!")
    End Try
  End Sub

  Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Try
con.Close()
con.Open()
cmd = New SqlCommand("update tournament set tname='" + TextBox2.Text + "',date='" +
TextBox3.Text + "',venue='" + TextBox4.Text + "',time='" + TextBox5.Text +
"',chiefguest='" + TextBox6.Text + "',description='" + TextBox7.Text + "' where tid='" +
TextBox1.Text + "'", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Tournament details Updated")
grid()
cleartxt()
con.Close()
    Catch ex As Exception
MessageBox.Show("Enter Valid Information for Updation!")
    End Try
  End Sub

  Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    If TextBox1.Text = "" Then
MessageBox.Show("Enter the Tournament ID")
    Else
        Try
con.Close()
con.Open()
            Dim cmd As New SqlCommand("delete from tournament where tid='" +
TextBox1.Text + "'", con)
```

```
cmd.ExecuteNonQuery()
MsgBox("Tournament details Deleted Successfully")
grid()
cleartxt()
con.Close()
            Catch ex As Exception
MessageBox.Show("Enter Valid Information for Deletion!")
            End Try
        End If
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
cleartxt()
    End Sub

    Private Sub GameDevelopmemtLevelToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
College.Show()
    End Sub

    Private Sub LogoutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
Login.Show()
    End Sub

    Private Sub TournamentRegistrationToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
TournamentRegistration.Show()
    End Sub

    Private Sub DateTimePicker1_ValueChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles DateTimePicker1.ValueChanged
        TextBox3.Text = DateTimePicker1.Value
    End Sub

    Private Sub TournamentResultToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
Result.Show()
    End Sub

    Private Sub Tournament_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
grid()
    End Sub
```

4. GAME DEVELOPMENT MODEL

```
Imports System.Data.SqlClient
Imports System.Data.Sql
Imports System.IO
Module PlayersList
   Public Sub Football(ByVal a)
con.Close()
con.Open()
      Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Foot Ball'", con)
      Dim da As New SqlDataAdapter(c)
      Dim ds As New DataSet
da.Fill(ds)
      a.DataSource = ds.Tables(0)
con.Close()
   End Sub
   Public Sub Cricket(ByVal a)
con.Close()
con.Open()
      Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Cricket'", con)
      Dim da As New SqlDataAdapter(c)
      Dim ds As New DataSet
da.Fill(ds)
      a.DataSource = ds.Tables(0)
con.Close()
   End Sub
   Public Sub Hockey(ByVal a)
con.Close()
con.Open()
      Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Hockey'", con)
      Dim da As New SqlDataAdapter(c)
      Dim ds As New DataSet
da.Fill(ds)
      a.DataSource = ds.Tables(0)
con.Close()
   End Sub
   Public Sub Volleyball(ByVal a)
con.Close()
con.Open()
      Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Volley Ball'", con)
      Dim da As New SqlDataAdapter(c)
      Dim ds As New DataSet
da.Fill(ds)
```

```
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Badminton(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Badminton'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Basketball(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Basket Ball'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Kabaadi(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Kabaadi'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Handball(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Hand Ball'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
```

```
    End Sub
    Public Sub Throwball(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Throw Ball'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Khokho(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Kho Kho'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Tabletennis(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Table Tennis'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Shuttle(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Shuttle'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Tennis(ByVal a)
```

```
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Tennis'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Gymnastics(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Gymnastics'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Weightliftting(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Weight Liftting'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Ballbadminton(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Ball Badminton'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Athletics(ByVal a)
con.Close()
con.Open()
```

```
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Athletics'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Halfmarathon(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Half Marathon'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Marathon(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Marathon'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Racewalk(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Race Walk'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub FMeterRelay(ByVal a)
con.Close()
con.Open()
```

```
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='4*100 Meter Relay'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub FFMeterRelay(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='4*400 Meter Relay'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Highjump(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='High Jump'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Longjump(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Long Jump'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Triplejump(ByVal a)
con.Close()
con.Open()
```

```
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Triple Jump'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Shotput(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Shot Put'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Discusthrow(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Discus Throw'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Hammerthrow(ByVal a)
con.Close()
con.Open()
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Hammer Throw'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Public Sub Javelinthrow(ByVal a)
con.Close()
con.Open()
```

```
        Dim c As New SqlCommand("select
address,sname,rno,class,department,gender,event,tournament,community,category from
student where event='Javelin Throw'", con)
        Dim da As New SqlDataAdapter(c)
        Dim ds As New DataSet
da.Fill(ds)
        a.DataSource = ds.Tables(0)
con.Close()
    End Sub
End Module
```

5. RESULT

```
Imports System.Data.SqlClient
Imports System.Data.Sql
Public Class Result
    Dim ss As Integer
    Dim cmd As New SqlCommand
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
 End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        If (TextBox1.Text = "" And TextBox2.Text = "" And TextBox3.Text = "" And
TextBox5.Text = "" And ComboBox1.Text = "" And ComboBox2.Text = "" And
TextBox6.Text = "") Then
MessageBox.Show("Please enter all values")
        Else
            Try
con.Close()
con.Open()
cmd = New SqlCommand("insert into result values('" + ComboBox1.Text + "','" +
TextBox1.Text + "','" + ComboBox2.Text + "','" + TextBox2.Text + "','" + TextBox3.Text +
"','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "')", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Result Save Successfully!")
grid()
cleartxt()
con.Close()
            Catch ex As Exception
MessageBox.Show("Enter Valid Input!")
            End Try
        End If
    End Sub
    Private Sub grid()
con.Close()
con.Open()
        Dim c As New SqlCommand("select * from result", con)
        Dim da As New SqlDataAdapter(c)
```

```
        Dim ds As New DataSet
da.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
con.Close()
    End Sub
    Private Sub cleartxt()
        ComboBox1.Text = ""
        ComboBox2.Text = ""
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim i As Integer
        i = InputBox("Enter Serial Number")
        Try
con.Close()
con.Open()
            Dim da As New SqlCommand("select * from result where sno='" + i.ToString() + "'",
con)
            Dim ds As SqlDataReader
ds = da.ExecuteReader
            While (ds.Read())
                ComboBox1.Text = ds.Item(0)
                TextBox1.Text = ds.Item(1)
                ComboBox2.Text = ds.Item(2)
                TextBox2.Text = ds.Item(3)
                TextBox3.Text = ds.Item(4)
                TextBox4.Text = ds.Item(5)
                TextBox5.Text = ds.Item(6)
                TextBox6.Text = ds.Item(7)
            End While
grid()
con.Close()
ds.Close()
        Catch ex As Exception
MessageBox.Show("Invalid Serial Number!")
        End Try
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Try
con.Close()
con.Open()
            cmd = New SqlCommand("update result set tid='" + ComboBox1.Text + "',tname='"
+ TextBox1.Text + "',sno='" + ComboBox2.Text + "',team='" + TextBox2.Text + "',sname='"
```

```
+ TextBox3.Text + "',event='" + TextBox4.Text + "',result1='" + TextBox5.Text + "',date='"
+ TextBox6.Text + "' where sno='" + ComboBox2.Text + "'", con)
cmd.ExecuteNonQuery()
MessageBox.Show("Tournament Result Details Updated")
grid()
cleartxt()
con.Close()
        Catch ex As Exception
MessageBox.Show("Enter Valid Information for Updation!")
        End Try
    End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If ComboBox2.Text = "" Then
MessageBox.Show("Enter the Serial Number")
        Else
            Try
con.Close()
con.Open()
                Dim cmd As New SqlCommand("delete from result where sno='" +
ComboBox2.Text + "'", con)
cmd.ExecuteNonQuery()
MsgBox("Tournament Result Details Deleted Successfully")
grid()
cleartxt()
con.Close()
            Catch ex As Exception
MessageBox.Show("Enter Valid Information for Deletion!")
            End Try
        End If
    End Sub
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
cleartxt()
    End Sub
Private Sub Result_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        TextBox6.Text = Now.Date
grid()
v1()
v2()
    End Sub
    Private Sub v1()
con.Close()
con.Open()
        Dim da As New SqlCommand("select tid from tregister", con)
        Dim ds As SqlDataReader
ds = da.ExecuteReader
        While (ds.Read())
ComboBox1.Items.Add(ds.Item(0))
```

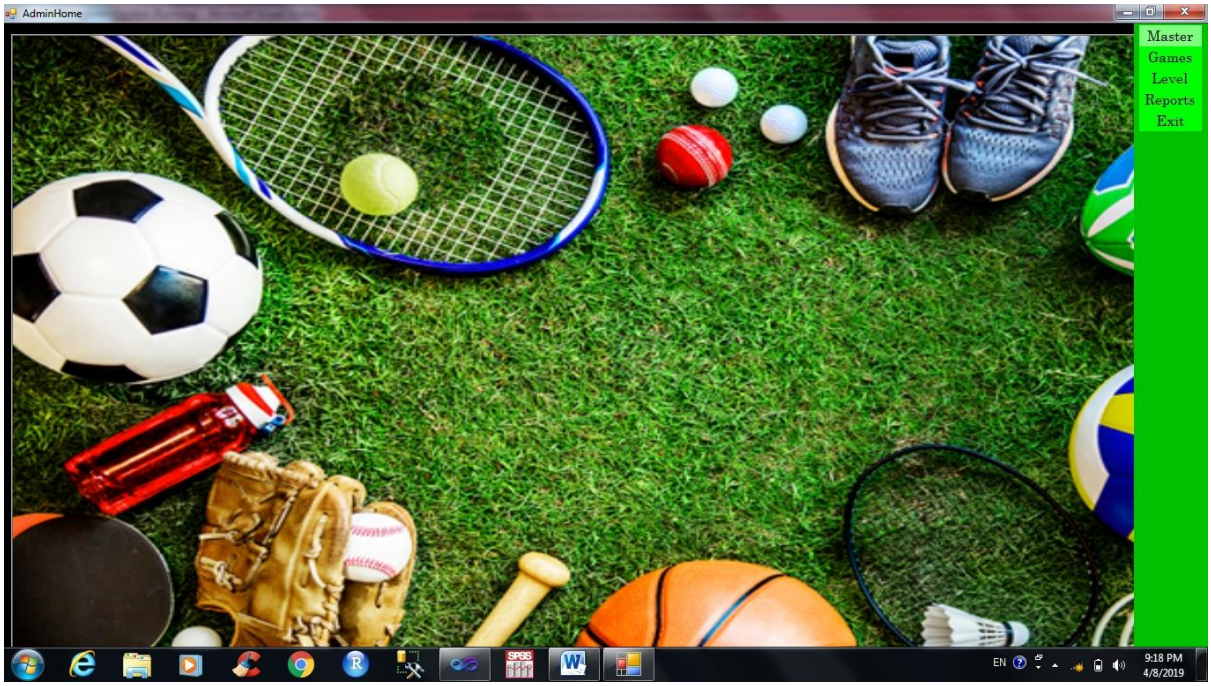
```
        End While
con.Close()
ds.Close()
    End Sub
    Private Sub v2()
con.Close()
con.Open()
        Dim da As New SqlCommand("select sno from tregister", con)
        Dim ds As SqlDataReader
ds = da.ExecuteReader
        While (ds.Read())
ComboBox2.Items.Add(ds.Item(0))
        End While
con.Close()
ds.Close()
    End Sub
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
Me.Close()
    End Sub
 Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
con.Close()
con.Open()
        Dim da As New SqlCommand("select tname from tregister where tid='" +
ComboBox1.Text + "'", con)
        Dim ds As SqlDataReader
ds = da.ExecuteReader
        While (ds.Read())
            TextBox1.Text = ds.Item(0)
        End While
con.Close()
ds.Close()
End Sub
Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox2.SelectedIndexChanged
con.Close()
con.Open()
        Dim da As New SqlCommand("select team,sname,event from tregister where sno='" +
ComboBox2.Text + "'", con)
        Dim ds As SqlDataReader
ds = da.ExecuteReader
        While (ds.Read())
            TextBox2.Text = ds.Item(0)
            TextBox3.Text = ds.Item(1)
            TextBox4.Text = ds.Item(2)
        End While
con.Close()
ds.Close()
    End Sub
```

E.SCREENSHOTS

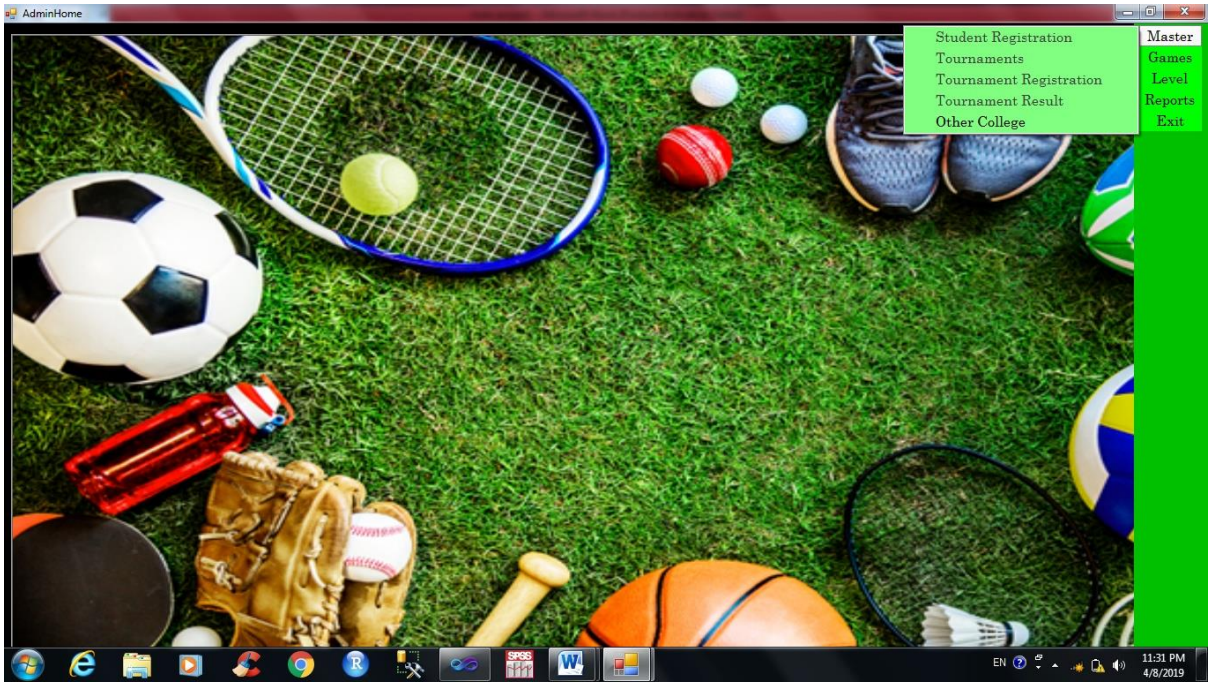
FRONT SCREEN

LOGIN FORM

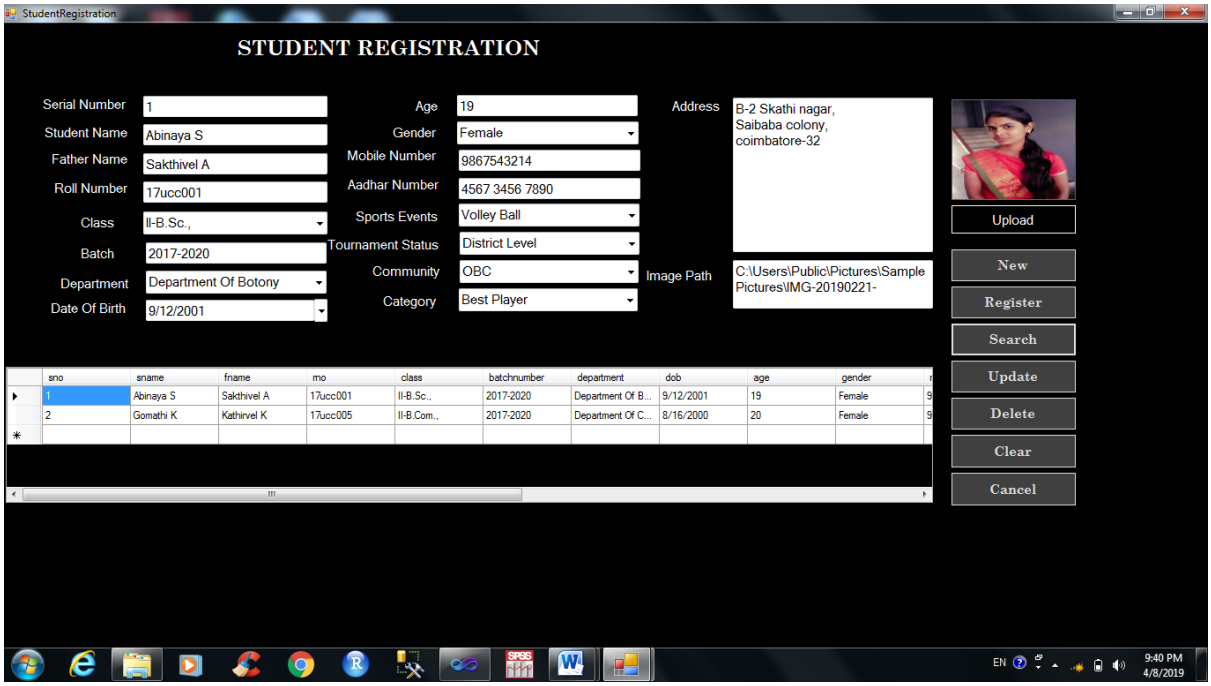
MASTER PAGE



MASTER PAGE FORM'S

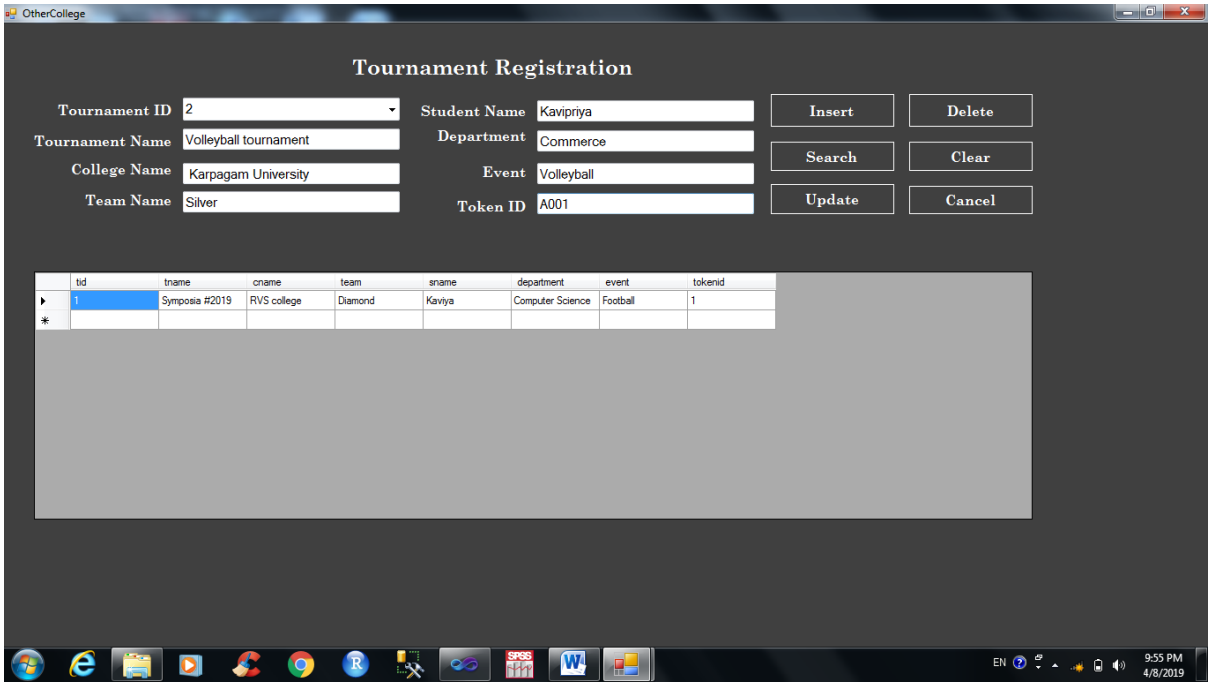


STUDENT REGISTRATION FORM



TOURNAMENT DETAILS FORM

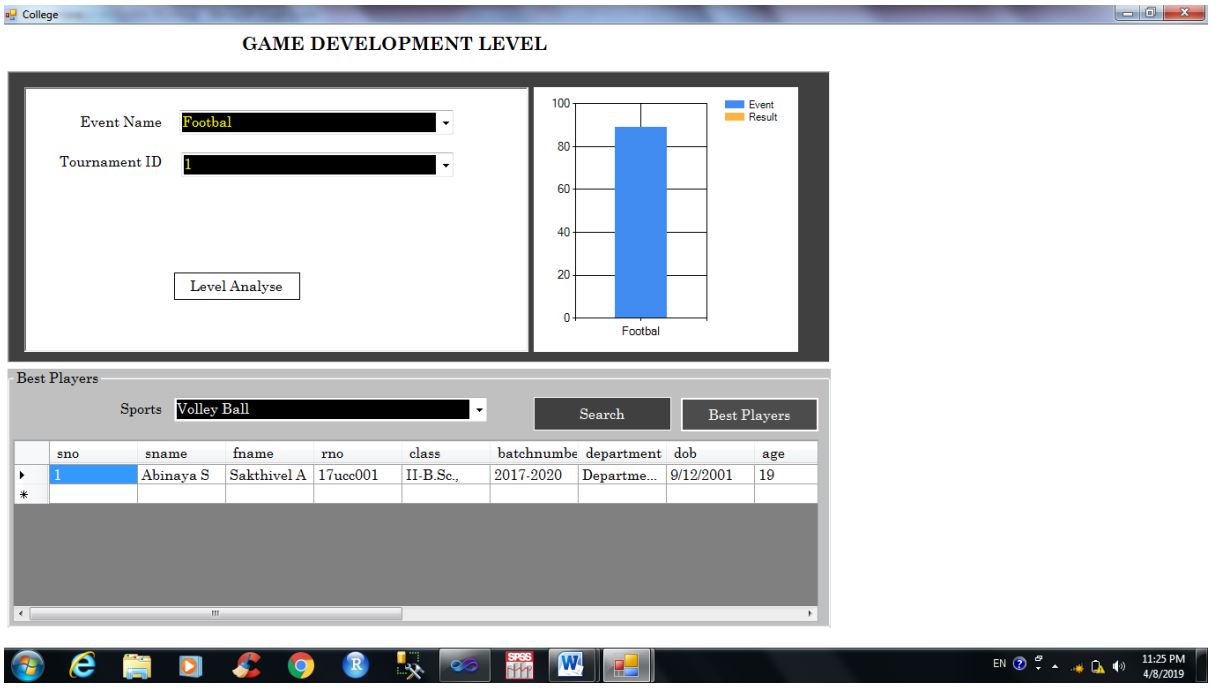
TOURNAMENT REGISTRATION FORM



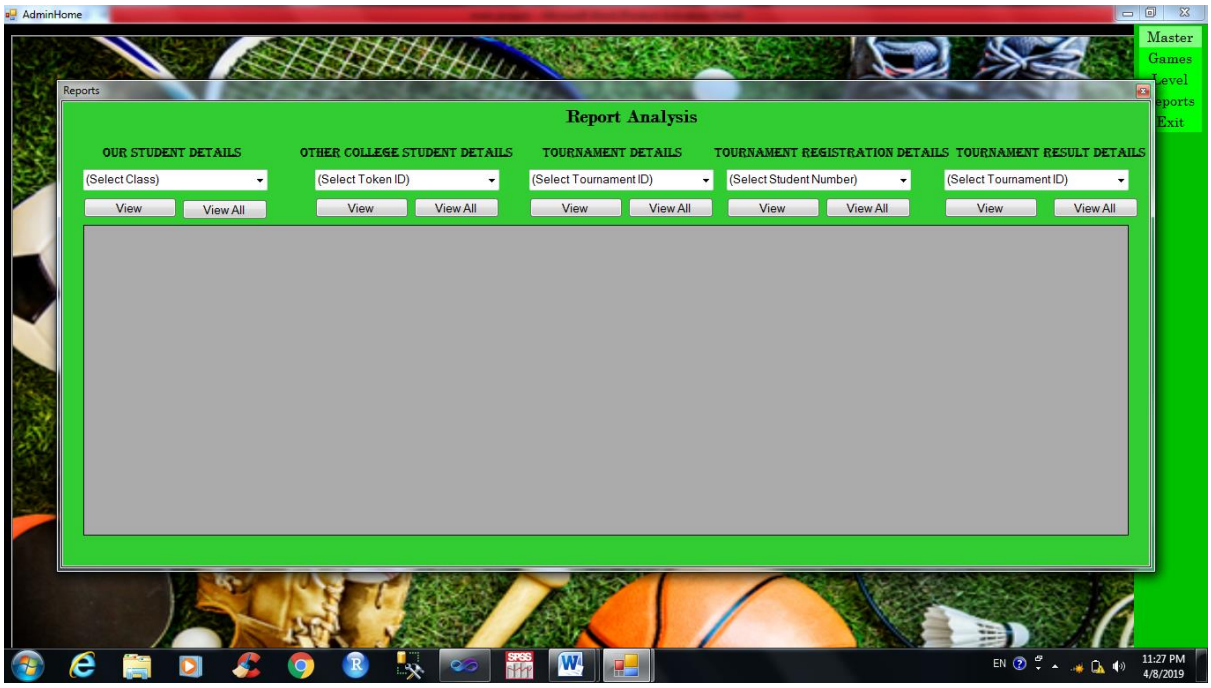
TOURNAMENT RESULT FORM

GAMES LIST

GAME DEVELOPMENT LEVEL FORM



REPORT FORM



STUDENT REPORT