KARNATAK UNIVERSITY

**Janata Shikshana Samiti's**

Shri Manjunatheshwara Institute of UG & PG Studies,
Vidyagiri, Dharwad-580004.

A PROJECT REPORT ON

# K.L.E SPORTS MANAGEMENT

BACHELOR OF COMPUTER APPLICATIONS (BCA)

OF

KARNATAK UNIVERSITY, DHARWAD

PROJECT GUIDED BY:

Miss.Medha Ghaligi

**Submitted by**

**Praveen Shetty(19U11201)**

**Varun Hegde(19U11262)**

**(Bachelor Of Computer Application) BCA VI SEMESTER**

**DEPARTMENT OF COMPUTER SCIENCE 2021-2022**

Janata Shikshana Samiti’s

Shri Manjunatheshwara Institute of UG & PG Studies,
Vidyagiri, Dharwad. 580004

# CERTIFICATE

**This is to certify that Mr.Praveen Shetty & Mr.Varun Hegde have satisfactorily completed Project work entitled “K.L.E SPORTS MANAGEMENT” for the partial fulfilment of BCA prescribed by Karnatak University, Dharwad during the academic year 2019-2020**

| | | |
|---|---|---|
| **Miss.Medha Ghaligi** | **Prof.Vivek.M.Laxmeshwar** | **Dr.Ajith Prasad** |
| **Project Guide** | **HOD** | **Principal** |

| **Examination Register No:** | **Examiners** |
|---|---|
| 1. **PRAVEEN SHETTY (19U11201)** | 1. ______________ |
| 2. **VARUN HEGDE(19U11262)** | 2. ______________ |

# ACKNOWLEDGEMENT

The Successful presentation of this project is acknowledgements of the immense support extends by JSS SMI UG & PG STUDIES , DHARWAD, which has provided an opportunity to fulfill the most cherished desired to reach our goal. We would like to express our heartfelt thanks to our President Shri Vishwaprasanna teertha Swamii of Pejavarmath, Udupi and Pujya Dr. D. Veerendra Heggade, the Chairman of Janata Shikshana Samiti & Dharmadhikari of Dharmastala and the Secretary of Janata Shikshana Samiti Dr.N.Vajrakumar. We would like to express our sincere and heartily thanks of gratitude to our beloved Principal Dr.Ajith Prasad and our Head of the computer department Shri. Vivek .M. Laxmeshwar who gave us the golden opportunity to do this wonderful project on the topic "K.L.E SPORTS MANAGEMENT", which also helped us in doing a lot of research and while doing so we were exposed to lot of new information which would help us in our mere future.

We would also take this opportunity to offer our sincere gratitude to our project guide Miss.Medha Ghaligi for her excellent support throughout the development of this project and providing the necessary information on our demand at all the time.

**Project Associates:**

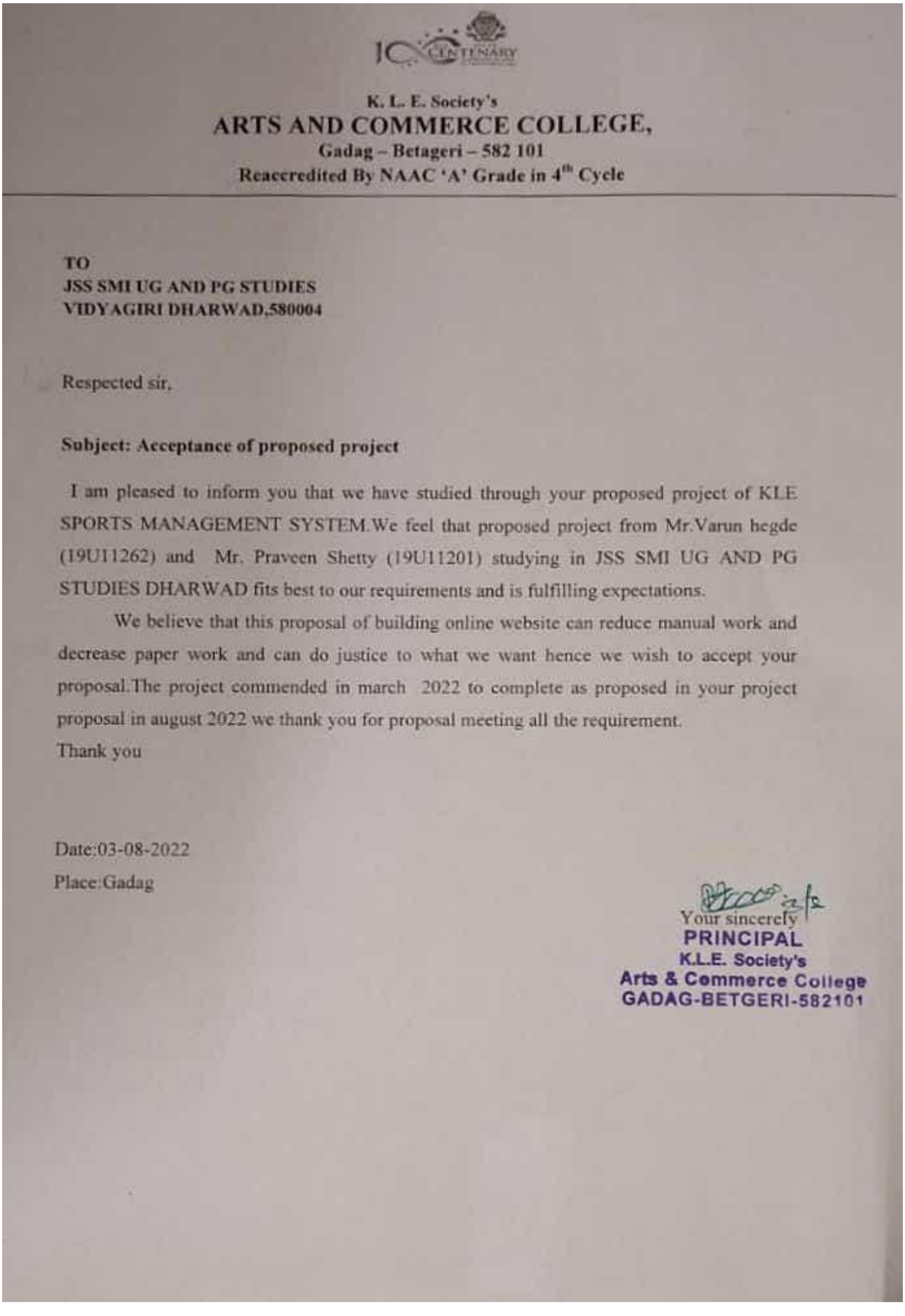
**Praveen Shetty**

**Varun Hegde**

# DECLARATION

We, **Praveen.Shetty and Varun.Hegde** student of sixth semester BCA, **Department of Computer Science, JSS SMI UG AND PG STUDIES VIDYAGIRI, DHARWAD, Karnatak University, Dharwad declare that the project entitled "K.L.E SPORTS MANAGEMENT" and submitted in partial fulfilment of the course requirement for the award of degree in Bachelor of Computer Application, Karnatak University, Dharwad during the academic year 2019-2020. We have not submitted the matter embodied to any other university or institution for the award of any other degree.**

| | |
|---|---|
| **Date:** | **Praveen.Shetty (19U11201)** |
| **Place: Dharwad** | **Varun.Hegde (19U11262)** |



K. L. E. Society's
ARTS AND COMMERCE COLLEGE,
Gadag – Betageri – 582 101
Reaccredited By NAAC 'A' Grade in 4th Cycle

**TO**
**JSS SMI UG AND PG STUDIES**
**VIDYAGIRI DHARWAD,580004**

Respected sir,

**Subject: Acceptance of proposed project**

I am pleased to inform you that we have studied through your proposed project of KLE SPORTS MANAGEMENT SYSTEM.We feel that proposed project from Mr.Varun hegde (19U11262) and Mr. Praveen Shetty (19U11201) studying in JSS SMI UG AND PG STUDIES DHARWAD fits best to our requirements and is fulfilling expectations.

We believe that this proposal of building online website can reduce manual work and decrease paper work and can do justice to what we want hence we wish to accept your proposal.The project commended in march 2022 to complete as proposed in your project proposal in august 2022 we thank you for proposal meeting all the requirement.

Thank you

Date:03-08-2022
Place:Gadag

Your sincerely
PRINCIPAL
K.L.E. Society's
Arts & Commerce College
GADAG-BETGERI-582101

# TABLE OF CONTENTS

# SYNOPSIS

## TITLE OF THE PROJECT

K.L.E SPORTS MANAGEMENT

## 1.1 INTRODUCTION

The web application, which facilitates the sports department to maintain the materials and stocks of the sports department. Managing the sports activities and maintaining the records of the department

.

## 1.2 OBJECTIVE OF THE PROJECT:

objective is to develop a web application for the sports department. To maintain the sports activities efficiently.

1) Event notification to students..
2) Get the availability of sports material requirements through web application and verify with the previous requests
3) Generate reports The for materials allocated.

## 1.3 Existing System

All the sports activities are maintained manually. The PD head has to send the circular to all departments to notify about the sports events.

### Limitations of Existing System

The existing system is a very time consuming process. Here all the sports activities are maintained manually. The PD head has to send the circular to all departments to notify about the sports events which is a manual process and time consuming process. The information may not reach all the student.

- Manual process
- Time consuming

- Announcing about their events become difficult
- Maintaining the records manually become difficult

## 1.4 Proposed System

- The web application, which facilitates the sports department to maintain the materials and stocks of the sports department.
- Managing the sports activities and maintaining the records of the department.
- With this system the physical department head can post the events like inter college and zonal events.
- The events posted by the department is available to students on the website

## 1.5 INPUT OF THE PROJECT

### Login Registration:

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.

### Student Registration:

The first procedure is the student registration. Here the employee enters all the details of a student including the sport they like to participate. All these information will be stored in the database.

### Event:

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

### Logout:

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

## 1.6 OUTPUT OF THE PROJECT

1. Sports management system is not only rewarding but it also helps the programmer to quickly organize the sports events interval of the time.
2. It will be able to check anything related to sports at any time.
3. Paper work and manual work is reduced through this system.
4. The system is user friendly and easy to use.

## 1.7 PROCESS LOGIC

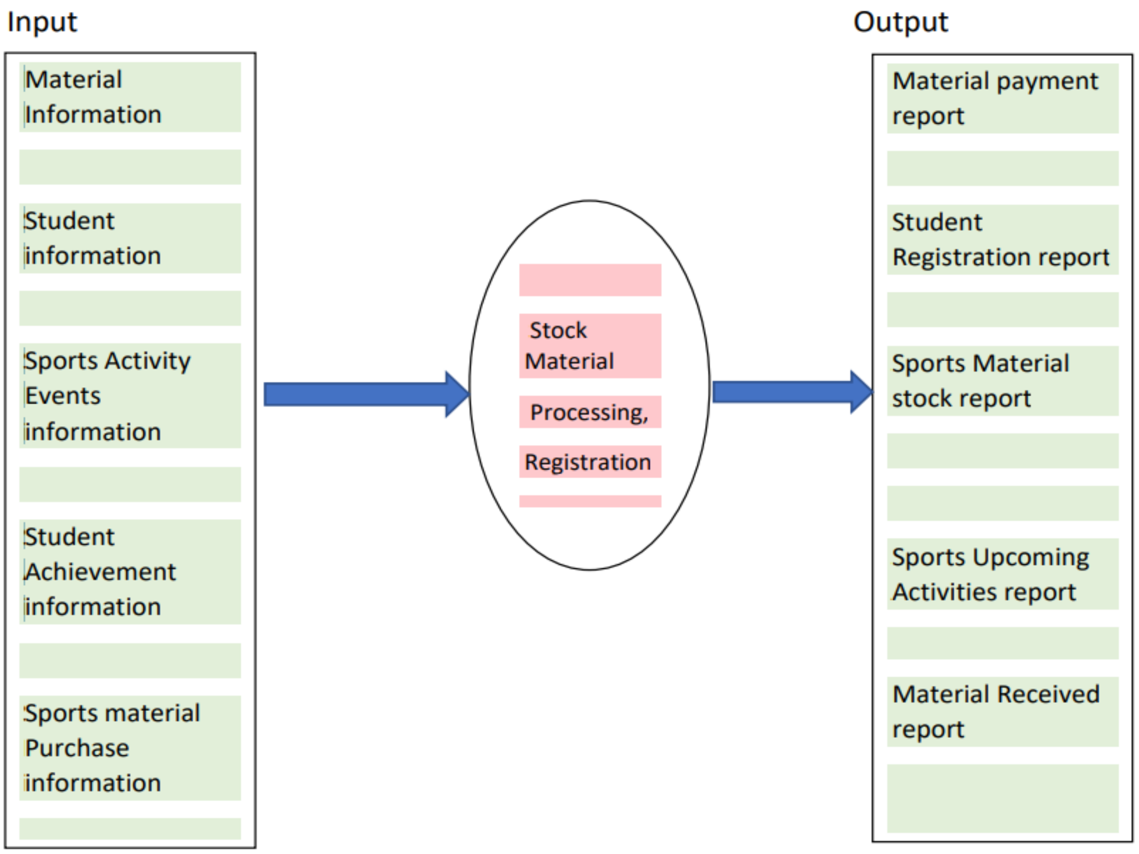
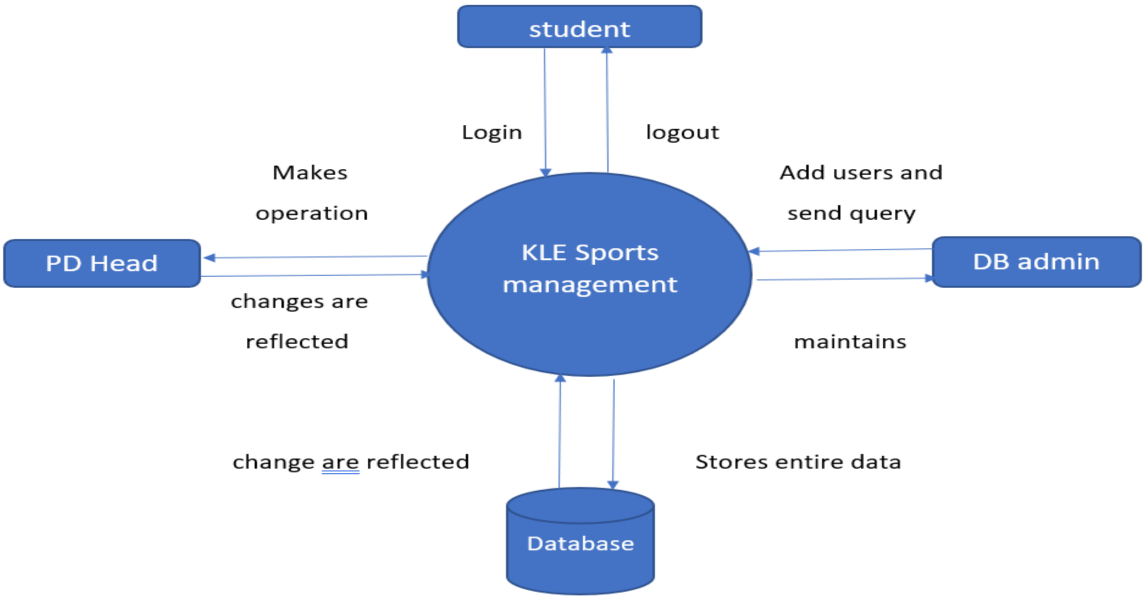

Input
Material Information
Student information
Sports Activity Events information
Student Achievement information
Sports material Purchase information
Stock Material
Processing,
Registration
Output
Material payment report
Student Registration report
Sports Material stock report
Sports Upcoming Activities report
Material Received report

**Figure 1.8 PROCESS LOGIC**

## Zero Level Data Flow Diagram



# 1.7 TOOLS/PLATFORM, LANGUAGES TO BE USED

## HARDWARE  REQUIREMENTS

| | |
|---|---|
| □ Hard disk | : 500GB or above |
| □ Processor | : Core i3 or above |
| □ RAM | : 2GB or above |

## SOFTWARE REQUIREMENTS

| | |
|---|---|
| □ Operating Environment | : Windows 7 or above |
| □ Front End | : HTML, CSS, Bootstrap |
| □ Middleware | : PHP |
| □ Backend | : MYSQL |
| □ Server Side Software | : XAMPP |

## 1.8 ARE YOU DOING THIS PROJECT FOR ANY INDUSTRY/CLIENT?
## IF YES ACCEPTANCE OF THE PROJECT.

- **YES**

## 1.9 DURATION OF THE PROJECT.

- **2 MONTHS**

## 1.10 MEMBERS OF THE PROJECT.

**□ 2 MEMBERS**

## 1.11 LIMITATIONS OF THE PROJECT.

- Time consuming.
- To keep database up to date regular update is needed

## 1.12 SCOPE OR FUTURE APPLICATION.

- Major operations that are done manually can be done within a matter of seconds.
- Live video stream and live score can be made

# Frame Work

## XAMPP ARCHITECTURE

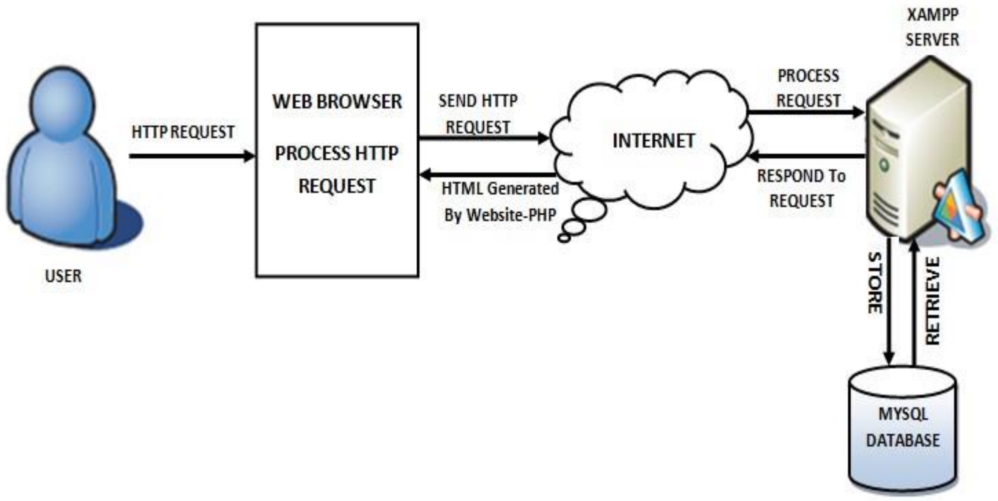


**FIGURE 1: XAMPP ARCHITECTURE**

**XAMPP** stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy.

## 2.1 What’s included in XAMPP?

XAMPP has three primary components. These are:

**2.1.1. Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.

**2.1.2 MySQL:** Every web application, however simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world‗s most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress, laravel etc.

## 2.2 Hypertext Pre-Processor (PHP):

PHP stands for Hypertext Preprocessor created by **RasmusLerdorf**, It is a serverside scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. However, PHP alone isn't enough in order to build dynamic web sites. To use PHP on a web site, we need a server that can process PHP scripts. XAMPP server allows developers to test PHP scripts locally, this makes it an invaluable piece of your local development environment.

Additionally, dynamic websites are dependent on stored information that can be added and updated easily, this is the main difference between a dynamic application and a static application. However, PHP doesn‗t provide a simple, efficient way to store data. This is where a relational database management system like MySQL comes into play.

**Syntax:**

```
<?php
//PHP CODE
?>
```

PHP originally stands for ―Hypertext Pre-Processor‖ or ―Personal Home Page‖ and was released as a free, open source project. Over time, the language was reworked to meet the needs of its users. In 1997, PHP was renamed to the current ―PHP: Hypertext

Pre-processor‖. PHP is generally used as a server-side scripting language; it is especially well suited for creating dynamic web pages and client-side GUI applications. PHP generally runs on a web server, taking PHP code as its input and creating web pages as output The scripting language features integrated support for interfacing with databases such as MySQL, which makes it a prime candidate for building all manner of web applications, from simple personal web sites to complex enterprise-level applications.

### 2.2.1 Usage:

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP‘s principal focus is server side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft‘s Active Server Pages, Sun Microsystems‘ Java Server Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include Cake PHP, Symfony, Code Igniter, and Zend Framework, offering features similar to other web application frameworks.

### 2.2.2 Speed Optimization:

As with many scripting languages, PHP scripts are normally kept as human-readable source code, even on production web servers. In this case, PHP scripts will be compiled at runtime by the PHP engine, which increases their execution time. PHP scripts are able to be compiled before runtime using PHP compilers as with other programming languages such as C (the language PHP and its extensions are written in). Code optimizers aim to reduce the computational complexity of the compiled code by reducing its size and making other changes that can reduce the execution time with the overall goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the Zend Optimizer PHP extension.

Another approach for reducing overhead for high load PHP servers is using PHP accelerators. These can offer significant performance gains by caching the compiled form of a PHP script in shared memory to avoid the overhead of parsing and compiling the code every time the script runs.

## 2.3 HTML:

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML

4.01, and this version was published in 2012.

**Syntax:**

```
<! DOCTYPE html>
<html>
<head>
<title>This is a Title </title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

**<! DOCTYPE>**

This tag defines the document type and HTML version.

**<HTML>**

This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.

**<HEAD>**

This tag represents the document's header which can keep other HTML tags like

<title> , <link> etc.

**<TITLE>**

The <title> tag is used inside the <head> tag to mention the document title.

**<BODY>**

This tag represents the document's body which keeps other HTML tags like<h1>, <div>, <p> etc.

HTML Elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as<img/> and <input/> introduce content into the page directly. Others such as<p>…. </p>, surround and provide information about document text and may include other tags as subelements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

## 2.4 Cascading Style Sheet (CSS):

**Cascading Style Sheets** was invented by **HakonWium Lie** on October 10, 1994 and maintained through a group of people within the W3C called the CSS Working Group CSSis the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. There are three types CSS mainly

- External CSS
- Internal CSS
- Inline CSS

**External** style sheets are separate files full of CSS instructions (with the file extension .css). When any web page includes an external style sheet, its look and feel will be controlled by this CSS file. This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page.

**Internal** styles are placed at the top of each web page document, before any of the content is listed. This is the next best thing to external, because they're easy to find, yet allow you to 'override' an external style sheet -- for that special page that wants to be a nonconformist.

**Inline** styles are placed right where you need them, next to the text or graphic you wish to decorate. You can insert inline styles anywhere in the middle of your

HTML code, giving you real freedom to specify each web page element. On the other hand, this can make maintaining web pages a real chore.

## Advantages of CSS

- **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it‘s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** − CSS can store web applications locally with the help of an offline cache.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

## 2.5 MYSQL:

SQL is a Relational Database Management System (RDBMS) that runs exclusively under the Windows operating system. One benefit of using Windows exclusively is that we can send and receive E-mail messages based on SQL "events" and we can also let the operating system handle login security. The data base is an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or a flexible manner. It includes facilities to add, modify or delete data from the SQL Server provides we with the software tools we need to organize that data in database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents. **Database Connection**

```
<?php
$servername ="localhost"; $username = "username"; $password =
"password"; $dbname = "database"; // Create connection $conn =new
mysqli($servername, $username, $password,$dbname); // Check connection
if (!$conn)  { die("Connection failed: " . $conn->connect_error());
}
?>
```

## 2.5.1 Features of SQL:

- It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc.
- It is not having condition loops, variables and most of the commands are single line commands.
- To implement application logics, SQL has got extension language popularly called as PL/SQL (Procedural language of sql).
- The entire SQL has been divided into 4 major categories.

✓ Data Manipulation Language.

✓ Data Definition Language.

✓ Transaction Control language.

✓ Data Control Language.

### 2.5.2 Security:

View are basically used as a part of security, means in many organizations ,the end user will never be given original tables & all data entry will be done with the help of views only. But the database administrator will be able to see everything because all the operations done by the different users will come to the same table.

# PROJECT SUBJECT

## Login Registration:

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.

## Student Registration:

The first procedure is the student registration. Here the employee enters all the details of a student including the sport they like to participate. All these information will be stored in the database.

## Event:

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

## Material Issue and Request:

Student can request to physical director for sports equipment and physical director can issue the sports equipment.It also maintains material return status.

## Event Request:

Student can apply for an event and physical director can accept or reject the student and their acceptance or rejection status is shown.

## Logout:

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

# SYSTEM REQUIREMENT & ANALYSIS

## 4.1 Definition

System Analysis is the detailed study of the various operations performed by the system and their relationships within and outside the system. Analysis is the process of breaking something into its parts so that the whole may be understood. System analysis is concerned with becoming aware of the problem, identifying the relevant and most decisional variables, analyzing and synthesizing the various factors and determining an optional or at least a satisfactory solution. During this a problem is identified, alternate system solutions are studied and recommendations are made about committing the resources used to the system.

## 4.2 Feasibility Study

A feasibility analysis usually involves a thorough assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go ahead with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were

- Technical Feasibility

- Economic Feasibility
- Behavioral Feasibility

### 4.2.1 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc, in order to give an introduction of technical system. Considering our project it is technically feasible.

### 4.2.2 Economic Feasibility

This feasibility study present tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

Thus feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy, timeliness.
- Cost comparison
- Estimate on the life expectancy of the hardware.
- Overall objective.

Our project is economically feasible. It does not require much cost to be involved in the overall process. The overall objective is in easing out the recruitment processes.

### 4.2.3 Behavioral / Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change.

## 4.3 Functional Requirements

Functional requirements are the statements of services the system should provide, how the system should respond the particular inputs and how the system should behave in particular situations.

**Users**

- There are two users for this application PD user and DB admin
- DB admin can perform all the valid operations like add, delete, modify users and materials

PD Head verifies students who registers

**Events by PD Head**

- PD Head can post any tournaments.
- Any student can request for conducting tournaments.
- Student's requests will be validated.

**Events by Students**

- Students can request materials for events conduction.
- PD Head accepts the event requests

**Material management**

- All available materials count will be kept.
- Materials damaged will be kept.

## 4.4 Non-Functional Requirements

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

### 4.4.1 Performance Requirements

- Performance: The application takes minimum 2-3 seconds to validate the information or execute the query.
- Reliability: Since the validation is done on each stage the information is processed in a correct and accurate form.
- Safety: The backup of database is always maintained in a local server. & information is securely transferred using https(Hyper-Text Transfer Protocol).

### 4.4.2 Software Quality Attributes

- Maintainability: Since each model has its own requirements. Therefore changes made on the module will not affect the other.
- Portability: Presently the application is compatible with windows 7, 8.1, 10 operating system.

### 4.4.3 Business Attributes

- **Cost**

In the development of the system the cost attribute requires the factors like hardware and software resources, specific amount of man power. As Windows is open source so it's not much cost effective.

- **Quality**

This software is properly normalized to avoid the redundancy and Ambiguity and hence, all the irrelevant constraints are removed.

### 4.4.4 Development Attributes

- **Testability:** The design phase of the system development takes into account the acceptance criteria and the factors affecting the performance, to make the system easily testable. The system will give high performance in terms of speed and storage efficiency. The system will provide the accurate output especially in maintaining & updating the account information.

## 4.5 Requirement Specification

### 4.5.1 Hardware Requirements:

| Hardware component | Description |
|---|---|
| **Memory** | 2GB RAM and above |
| **Hard Disk** | 100GB and above |
| **Processor** | Core i3 and above |

### 4.5.2 Software Requirements:

| Software component | Description |
|---|---|
| **Operating System** | Windows 7 and above |
| **Scripting Languages** | Java Script |
| **Language** | PHP |
| **Database** | MYSQL |
| **Server** | Apache Tomcat Server |

# DATA FLOW DIAGRAMS (DFD)

A data flow diagram (DFD) is a significant modelling technique for analyzing and constructing information processes. Data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).On a DFD, data items flow from an external data source or an internal data store to an Internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart.

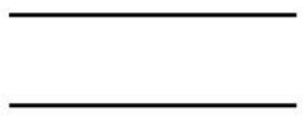
This shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD). Dataflow diagrams provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data-flow diagram.

With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. A designer usually draws a context-level DFD showing the relationship between the entities inside and outside of a system as one single step. This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modelled. Numerous levels may be required to explain a complicated system. The different versions are Context Diagrams (Level 0), Partitioned Diagrams (single process only -- one level), functionally decomposed, levelled sets of Data Flow Diagrams.

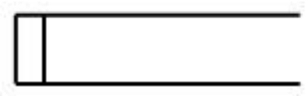
## 6.1 Data Flow Diagrams Symbols

A DFD usually comprises of four components. These four components can be represented by four simple symbols. These symbols can be explained in detail as follows: External entities (source/destination of data) are represented by squares; Processes (inputprocessing-output) are represented by rectangles with rounded corners; Data Flows (Physical or electronic data) are represented by arrows; and finally, Data Stores (physical or electronic like XML files) are represented by open-ended rectangles.

## 6.2 Data store

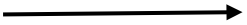


Or



A data store stores data passively for later access. A data store responds to requests to store and access data. It does not generate any operations. A data store allows values to be accessed in an order different from the order in which they were generated. Input flows indicate information or operations that modify the stored data such as adding or deleting elements or changing values. Output flows indicate information retrieved from the store; this information can be an entire value or a component of a value.

## 6.3 Data flow

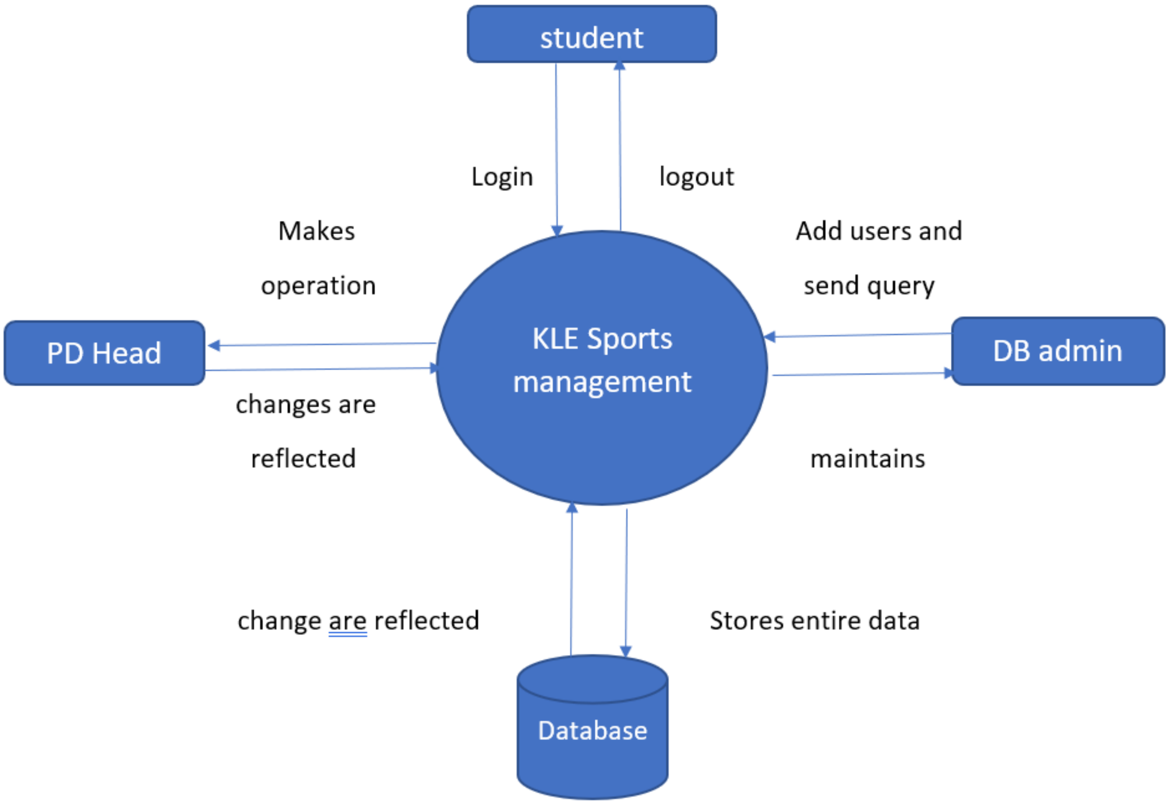


A data flow moves data between processes or between processes and data stores. As such, it represents a data value at some point within a computation and an intermediate value within a computation if the flow is internal to the diagram. This value is not changed. The names of input and output flows can indicate their roles in the computation or the type of the value they move. Data names are preferably nouns. The name of a typical piece of data, the data aspect, is written alongside the arrow.

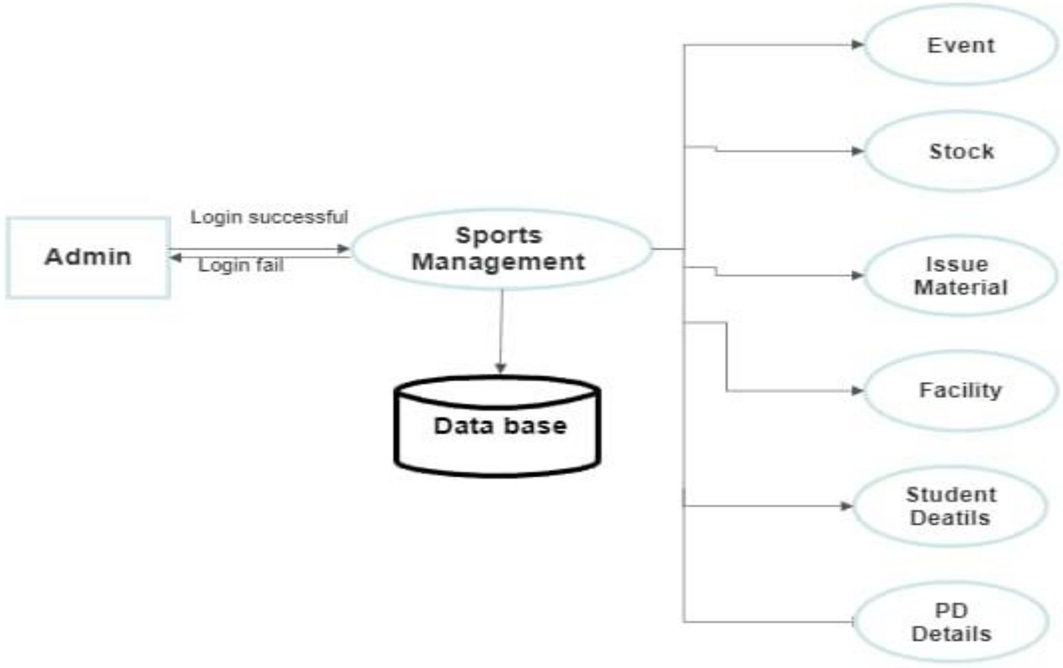
## 6.4 Rules for Creating DFD

- Processes should be named and numbered for easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source (upper left corner) to the destination although they may flow back to a source. One way to indicate this is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal in the lower right corner.
- When a process is exploded into the lower-level details, they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

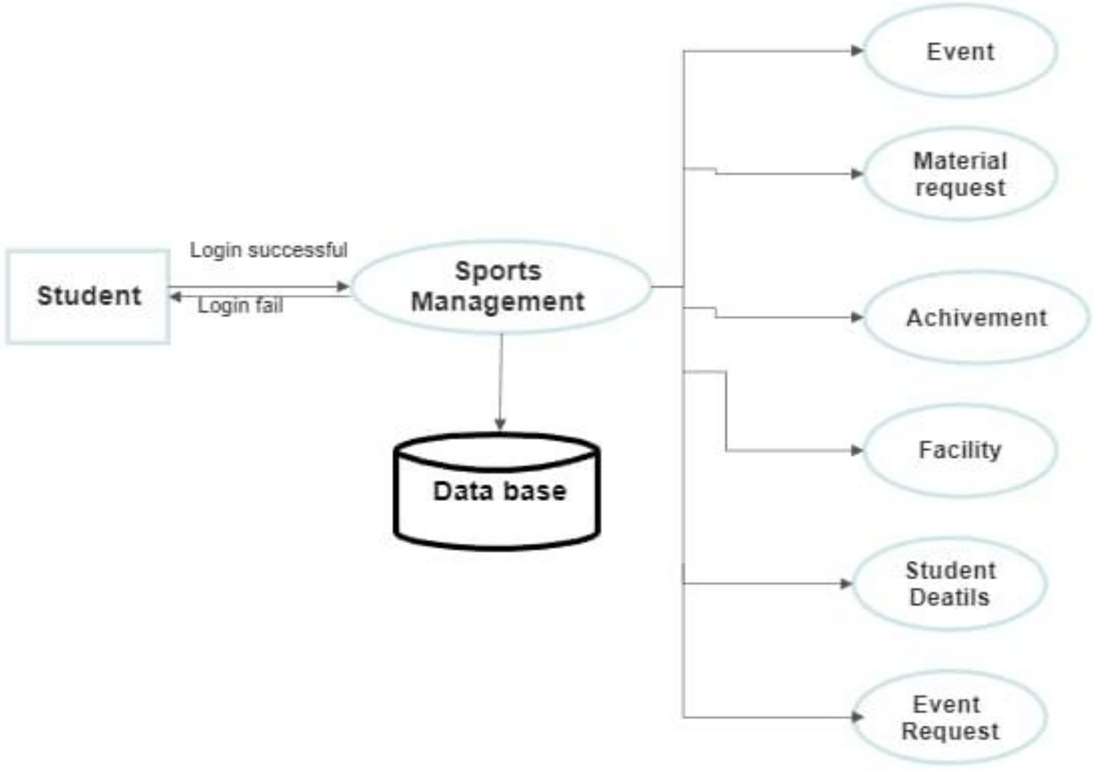
Process and data flow names have the first letter of each word capitalized. **6.5 Zero Level DFD**



## Admin Level 1 Data Flow Diagram



## Student Level 1 Data Flow Diagram



# ER DIAGRAM

An entity-relationship diagram is a data modelling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. An **entity-relationship model** (**ERM**) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called **entityrelationship diagrams**, **ER diagrams**, or **ERDs**.

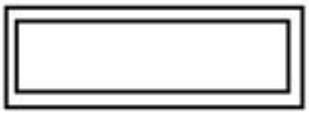
The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Sometimes, both of these phases are referred to as "physical design".

**The three main components of an E-R Diagram are:**

- The entity is a person, object, place or event for which data is collected. For example, if you consider the information system for a business, entities would include not only customers, but the customer‘s address, and orders as well. The entity is represented by a rectangle and labelled with a singular noun.
- The relationship is the interaction between the entities. In the example above, the customer places an order, so the word ―places‖ defines the relationship between that instance of a customer and the order or orders that they place. A relationship may be represented by a diamond shape, or more simply, by the line connecting the entities. In either case, verbs are used to label the relationships.
- The cardinality defines the relationship between the entities in terms of numbers. An entity may be optional: for example, a sales representative could have no

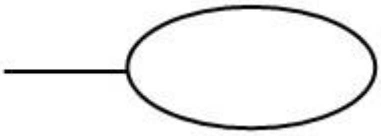
customers or could have one or many customers; or mandatory: for example, there must be at least one product listed in an order. There are several different types of cardinality notations; crow‘s foot notation, used here, is a common one. In crow‘s foot notation, a single bar indicates one, a double bar indicates one and only one (for example, a single instance of a product can only be stored in one warehouse), a circle indicates zero, and a crow's foot indicates many. The three main cardinal relationships are: oneto-one, expressed as 1:1; one-to-many, expressed as 1: N; and many-to-many, expressed as M: N.

## Weak Entity:



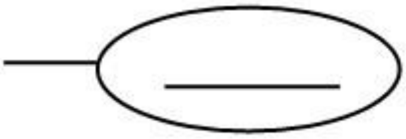
A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

## Attribute:



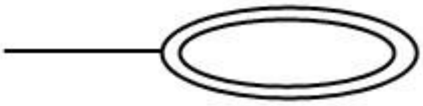
Each entity has attributes or particular properties that describe the entity. Most of the data in a database consists of values of attributes. The set of all possible values of an attribute is the attribute domain. In an ER model, an attribute name appears in an oval that has a line to the corresponding entity box.

## Key attributes:



A key attribute is the unique, distinguishing characteristic of the entity. An attribute or set of attributes that uniquely identifies a particular entity is a key. A key attribute in an ER Diagram is represented by an oval that has a line inside it and a line to the corresponding entity box. For example, an employee's social security number might be the employee's key attribute.

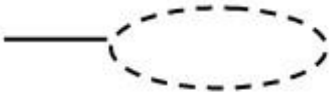
## Multi-valued attribute:



A multi-valued attribute can have more than one value. We indicate this with a double oval.

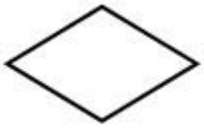
For example, an employee entity can have multiple skill values.

## Derived attribute:



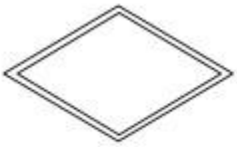
A derived attribute is based on another attribute. It is denoted by a oval and dotted line within it. For example, an employee's monthly salary is based on the employee's annual salary.

## Relationships:



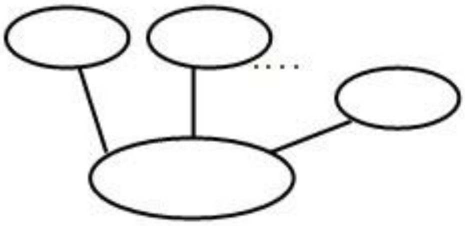
Relationships illustrate how two entities share information in the database structure. An association among entities is called a relationship. An attribute can also be a property of a relationship set. The association among the entities is described as one-to-one, one-to-many, many-to-many. A relationship is indicated by a rhombus.

## Identifying relationship:



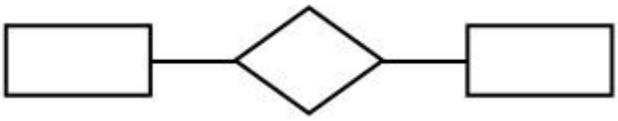
Identifying relationship is denoted by double rhombus.

## Composite Attribute



A composite attribute has multiple components and each component is atomic or composite. We illustrate this composite nature in the ER model by branching off the component attributes.

## Total Participation:

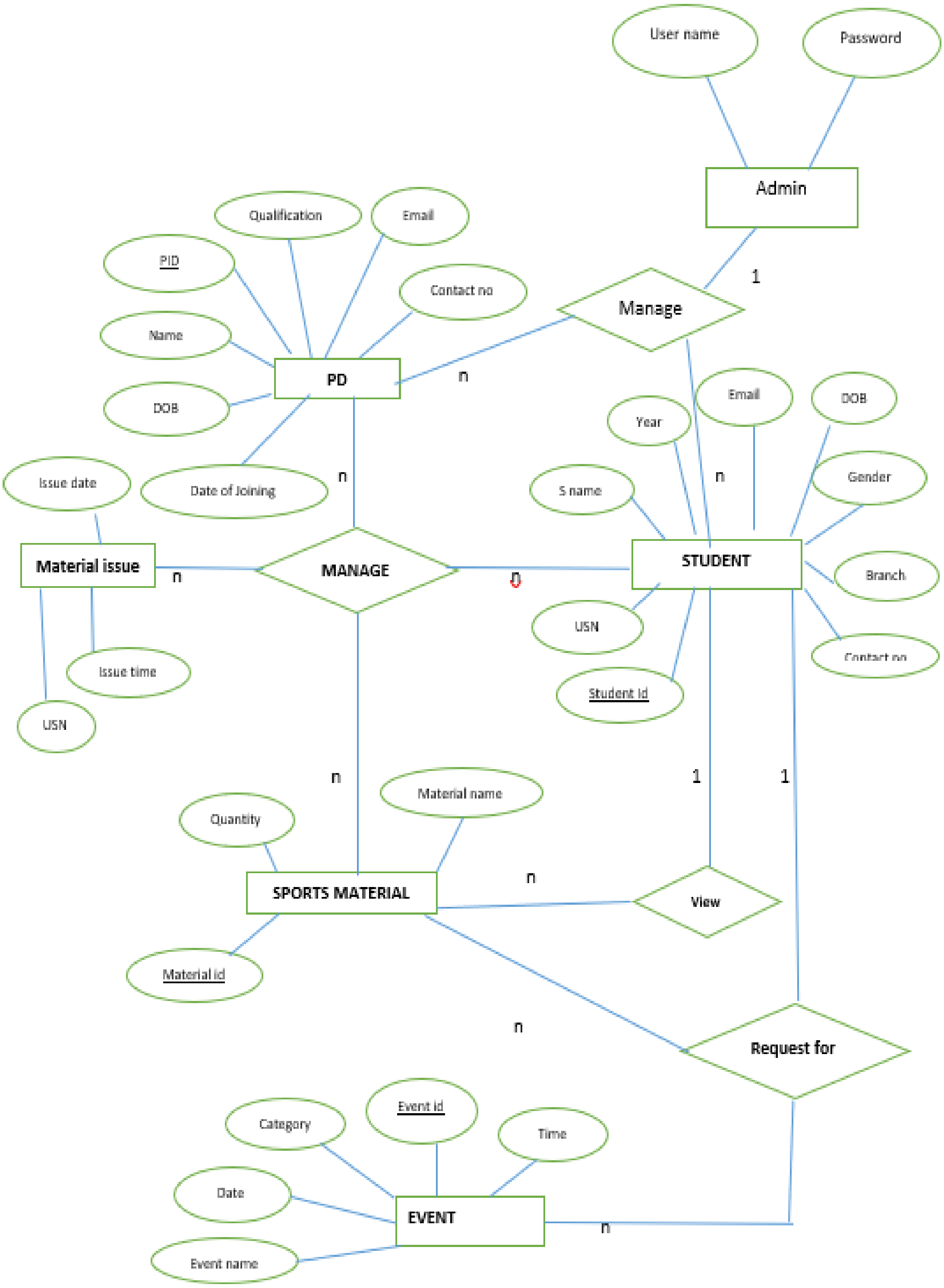


Total participation is represented by a double line.

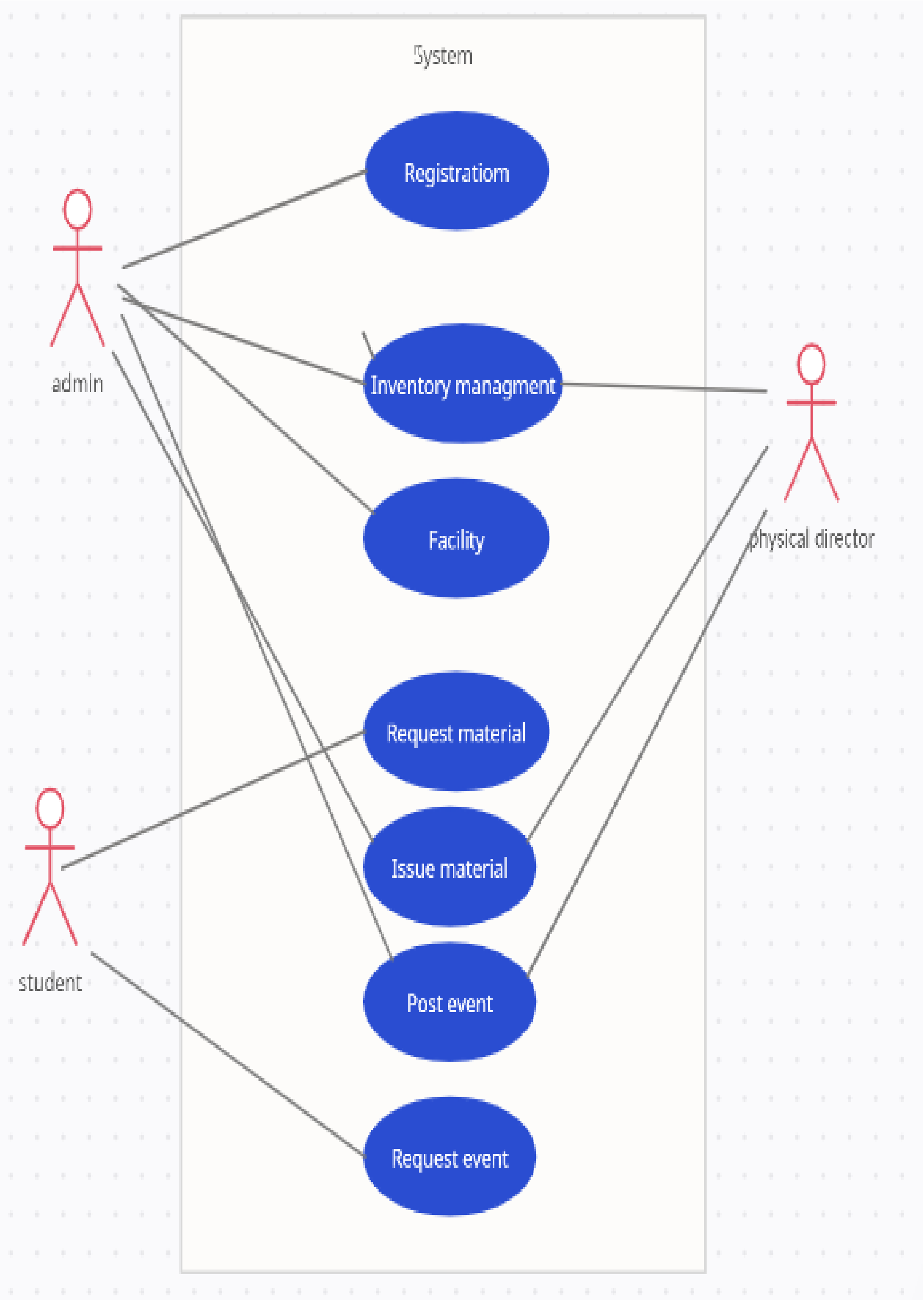
## Cardinality:

Cardinality specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinarily specifies the absolute minimum number of relationship.

## Entity Relationship Diagram

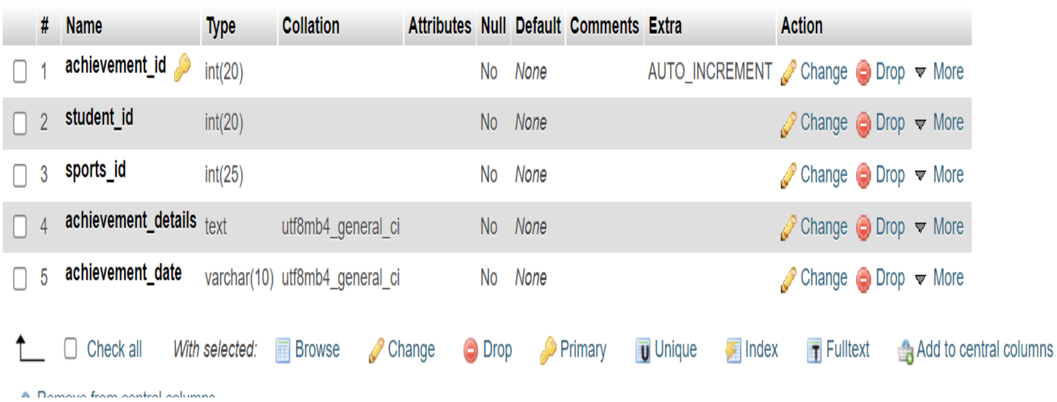


## Use case diagram



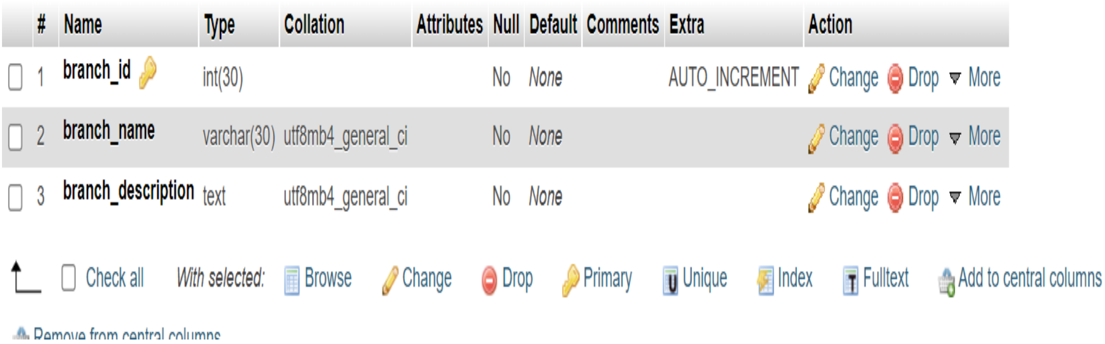
# TABLES USED IN DATABASE

## Achievement table:



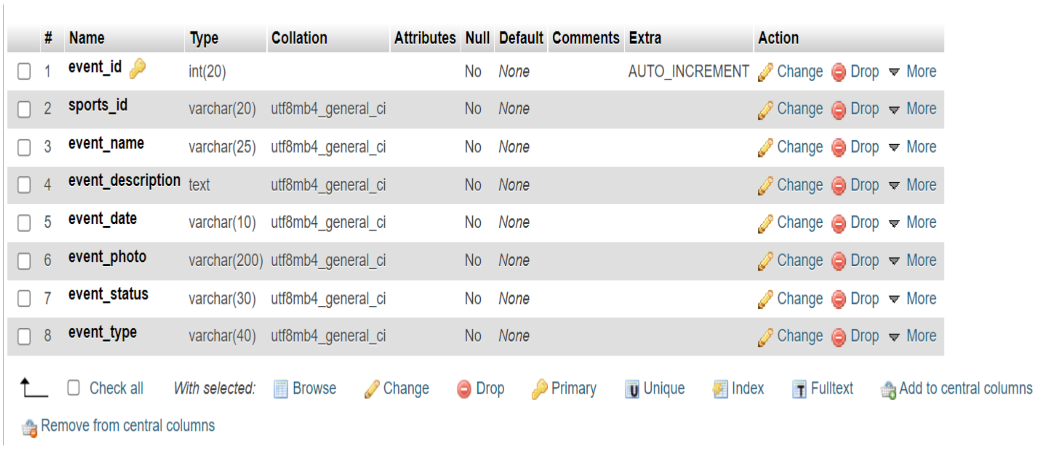
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | achievement_id | int(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop ▾ More |
| □ | 2 | student_id | int(20) | | | No | *None* | | | Change Drop ▾ More |
| □ | 3 | sports_id | int(25) | | | No | *None* | | | Change Drop ▾ More |
| □ | 4 | achievement_details | text | utf8mb4_general_ci | | No | *None* | | | Change Drop ▾ More |
| □ | 5 | achievement_date | varchar(10) | utf8mb4_general_ci | | No | *None* | | | Change Drop ▾ More |

## Branch table:



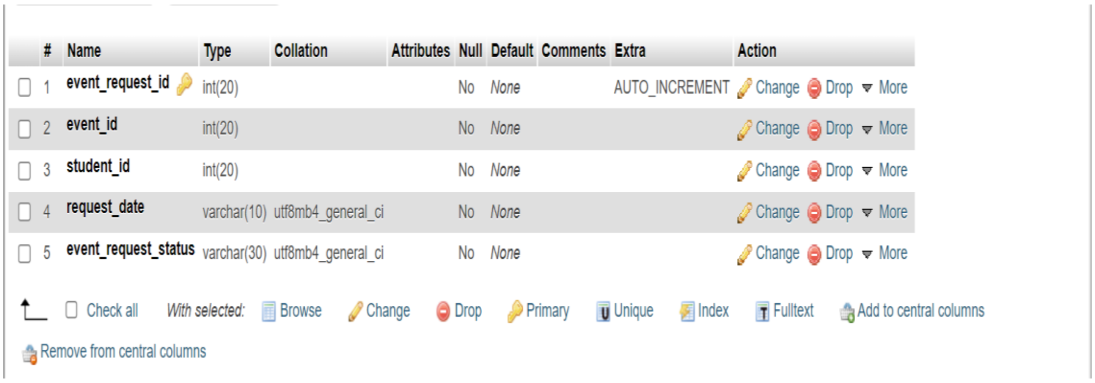
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | branch_id | int(30) | | | No | *None* | | AUTO_INCREMENT | Change Drop ▾ More |
| □ | 2 | branch_name | varchar(30) | utf8mb4_general_ci | | No | *None* | | | Change Drop ▾ More |
| □ | 3 | branch_description | text | utf8mb4_general_ci | | No | *None* | | | Change Drop ▾ More |

## Event table:



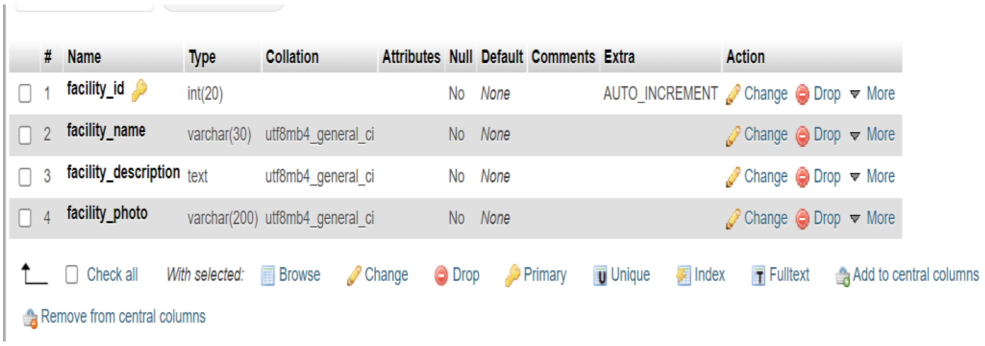
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | event_id | int(20) |  |  | No | None |  | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | sports_id | varchar(20) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 3 | event_name | varchar(25) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 4 | event_description | text | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 5 | event_date | varchar(10) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 6 | event_photo | varchar(200) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 7 | event_status | varchar(30) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 8 | event_type | varchar(40) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |

## Event_request table:



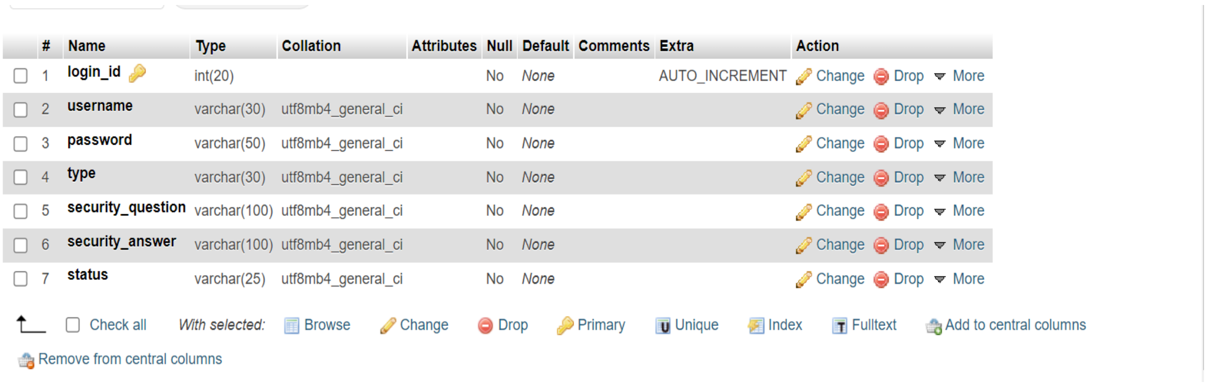
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | event_request_id | int(20) |  |  | No | None |  | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | event_id | int(20) |  |  | No | None |  |  | Change Drop More |
| ☐ | 3 | student_id | int(20) |  |  | No | None |  |  | Change Drop More |
| ☐ | 4 | request_date | varchar(10) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 5 | event_request_status | varchar(30) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |

## Facility table:



|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | facility_id | int(20) |  |  | No | None |  | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | facility_name | varchar(30) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 3 | facility_description | text | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |
| ☐ | 4 | facility_photo | varchar(200) | utf8mb4_general_ci |  | No | None |  |  | Change Drop More |

## Login table:

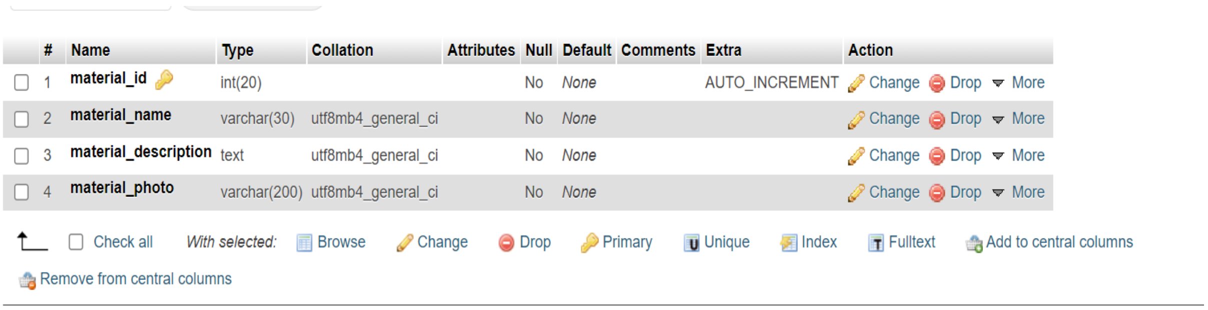


| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **login_id** 🔑 | int(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **username** | varchar(30) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 3 | **password** | varchar(50) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 4 | **type** | varchar(30) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 5 | **security_question** | varchar(100) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 6 | **security_answer** | varchar(100) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 7 | **status** | varchar(25) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |

☐ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

## Material table:

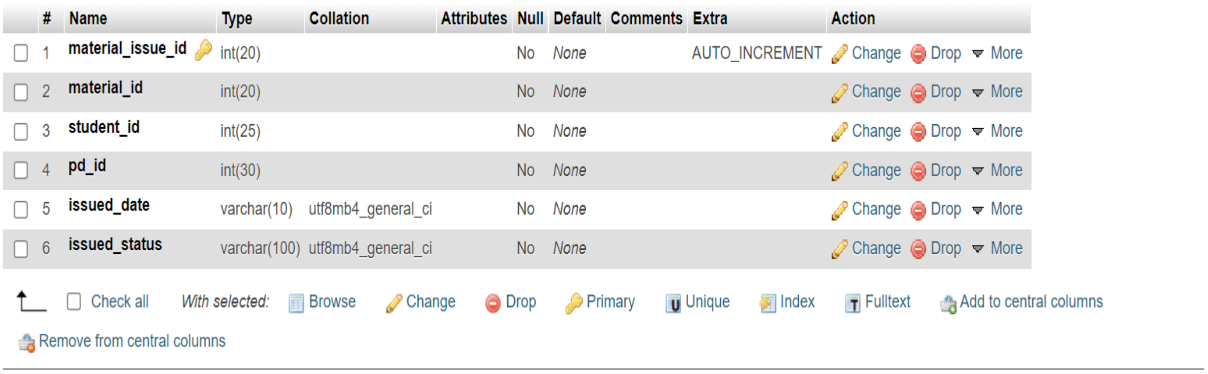


| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **material_id** 🔑 | int(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **material_name** | varchar(30) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 3 | **material_description** | text | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 4 | **material_photo** | varchar(200) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |

☐ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

## Material_issue table:

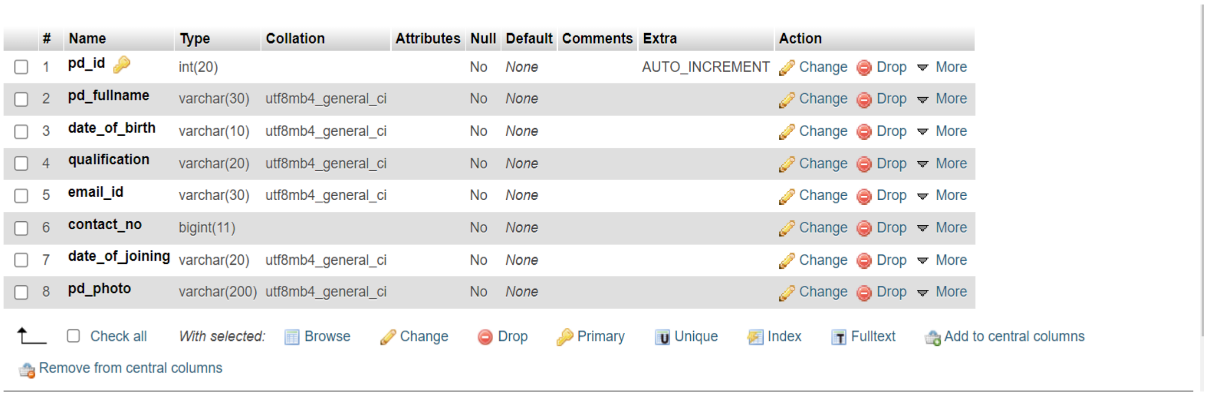


| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **material_issue_id** 🔑 | int(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **material_id** | int(20) | | | No | *None* | | | Change Drop More |
| ☐ | 3 | **student_id** | int(25) | | | No | *None* | | | Change Drop More |
| ☐ | 4 | **pd_id** | int(30) | | | No | *None* | | | Change Drop More |
| ☐ | 5 | **issued_date** | varchar(10) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 6 | **issued_status** | varchar(100) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |

☐ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

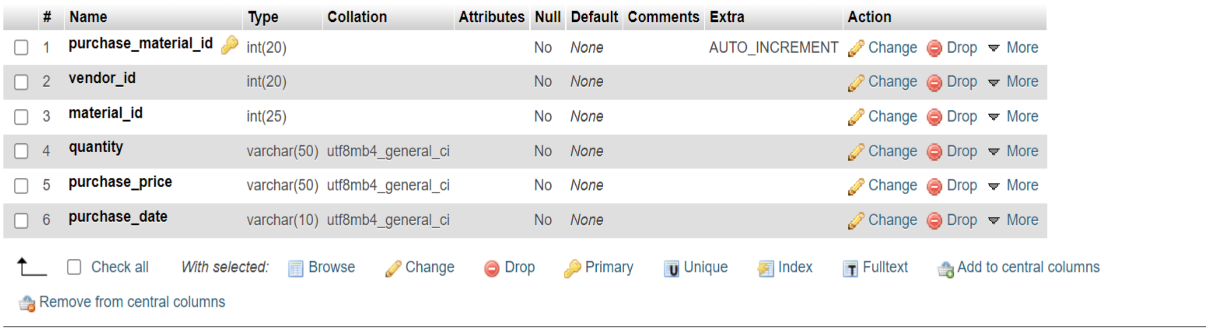
Remove from central columns

## Pd_details table:



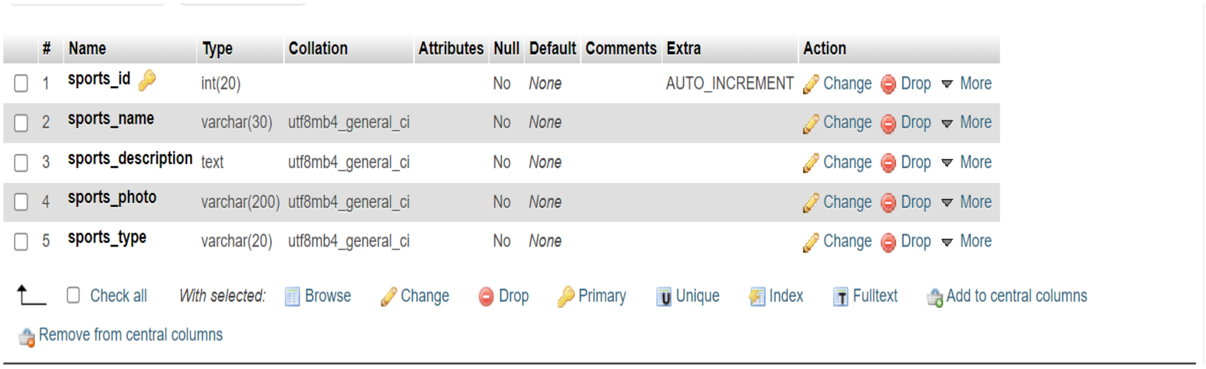
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | pd_id | int(20) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | pd_fullname | varchar(30) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 3 | date_of_birth | varchar(10) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 4 | qualification | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 5 | email_id | varchar(30) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 6 | contact_no | bigint(11) | | | No | None | | | Change Drop More |
| □ | 7 | date_of_joining | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 8 | pd_photo | varchar(200) | utf8mb4_general_ci | | No | None | | | Change Drop More |

## Purchase_material table:



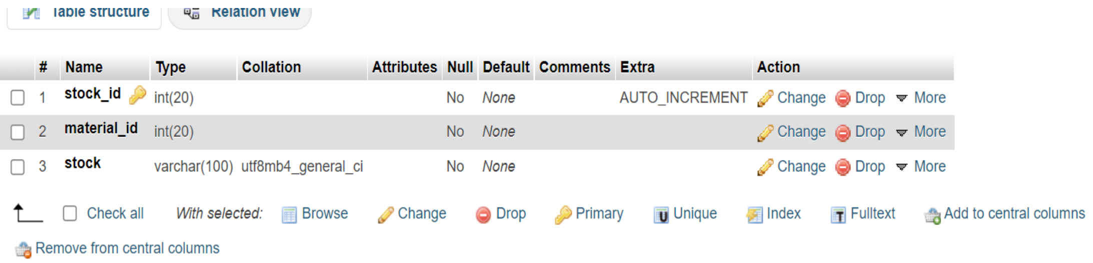
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | purchase_material_id | int(20) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | vendor_id | int(20) | | | No | None | | | Change Drop More |
| □ | 3 | material_id | int(25) | | | No | None | | | Change Drop More |
| □ | 4 | quantity | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 5 | purchase_price | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 6 | purchase_date | varchar(10) | utf8mb4_general_ci | | No | None | | | Change Drop More |

## Sports table:



| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | sports_id | int(20) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | sports_name | varchar(30) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 3 | sports_description | text | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 4 | sports_photo | varchar(200) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| □ | 5 | sports_type | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |

## Stock table:

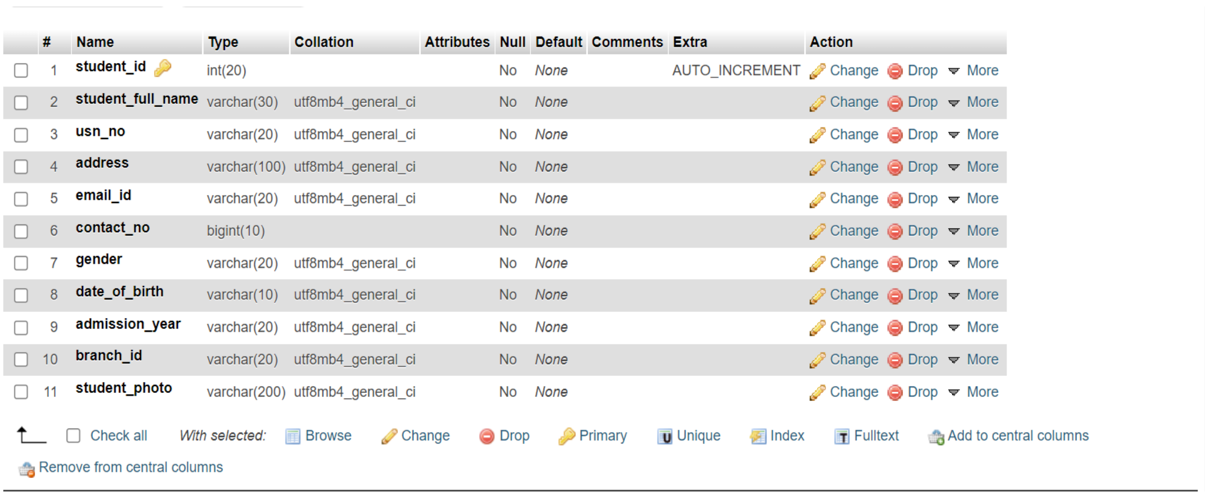


|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | stock_id | int(20) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | material_id | int(20) |  |  | No | *None* |  |  | Change Drop More |
| □ | 3 | stock | varchar(100) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

## Student table:

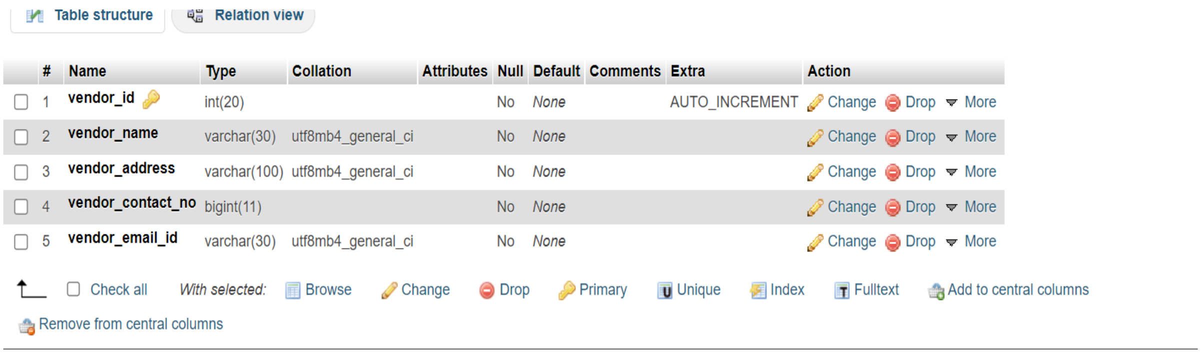


|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | student_id | int(20) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | student_full_name | varchar(30) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 3 | usn_no | varchar(20) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | address | varchar(100) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | email_id | varchar(20) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 6 | contact_no | bigint(10) |  |  | No | *None* |  |  | Change Drop More |
| □ | 7 | gender | varchar(20) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 8 | date_of_birth | varchar(10) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 9 | admission_year | varchar(20) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 10 | branch_id | varchar(20) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 11 | student_photo | varchar(200) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

## Vendor table:



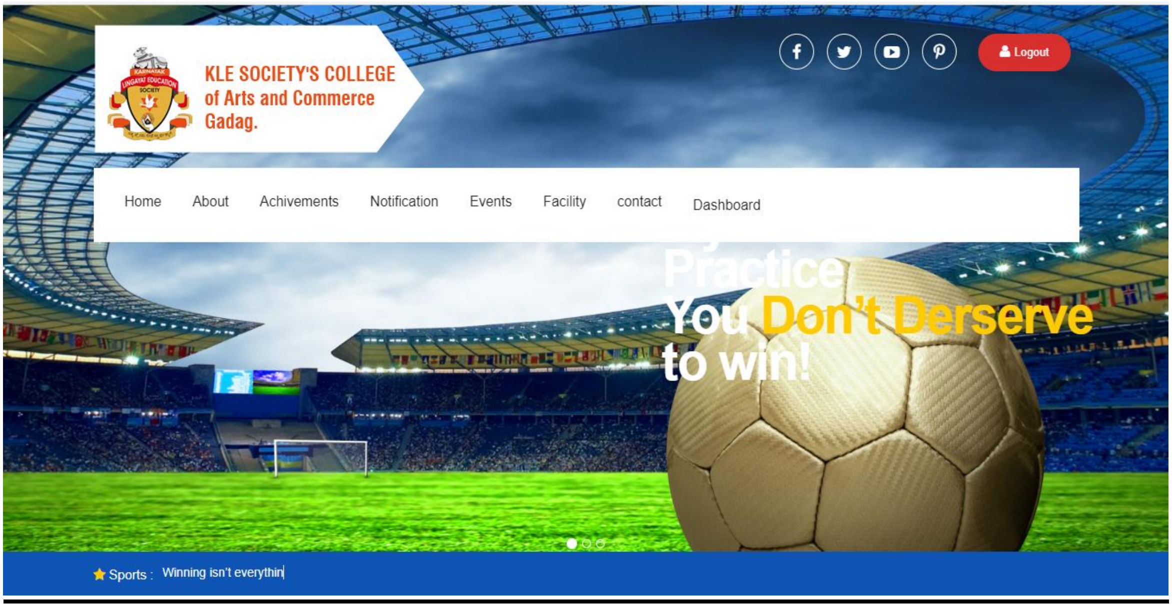
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | vendor_id | int(20) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | vendor_name | varchar(30) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 3 | vendor_address | varchar(100) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | vendor_contact_no | bigint(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 5 | vendor_email_id | varchar(30) | utf8mb4_general_ci |  | No | *None* |  |  | Change Drop More |

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

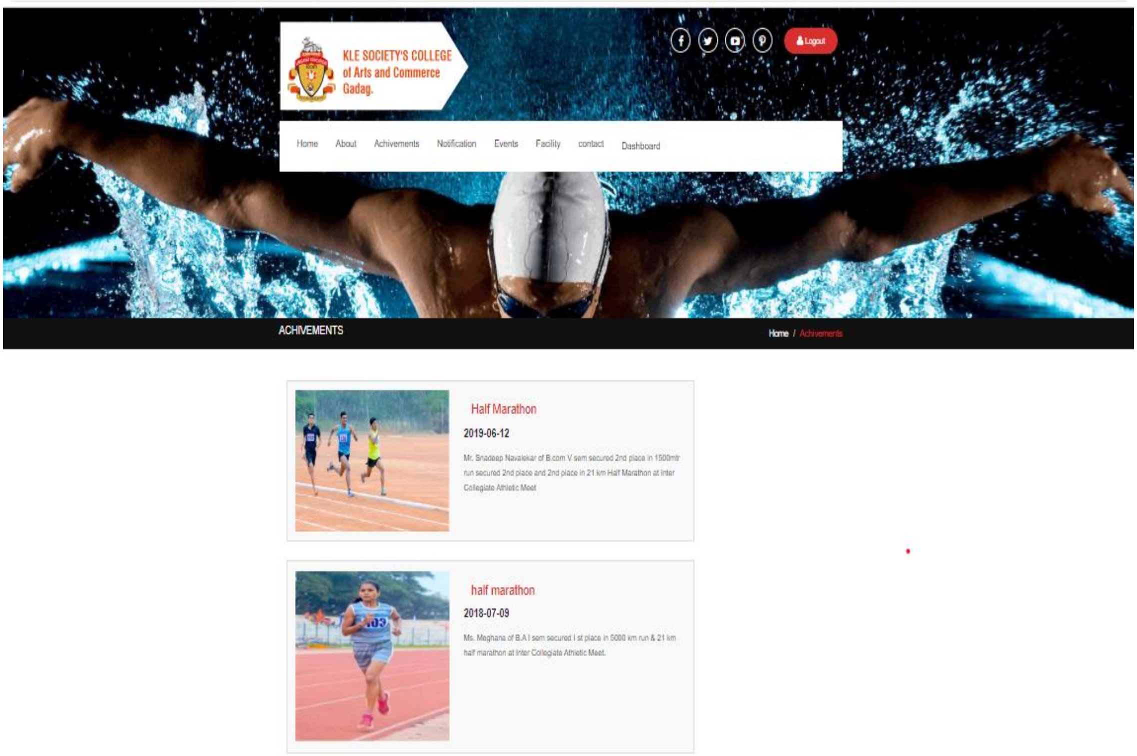
Remove from central columns

# SCREENSHOTS /SAMPLEOUTPUT

## MAIN HOME PAGE:



## Achievement page:



ACHIVEMENTS

Half Marathon

2019-06-12

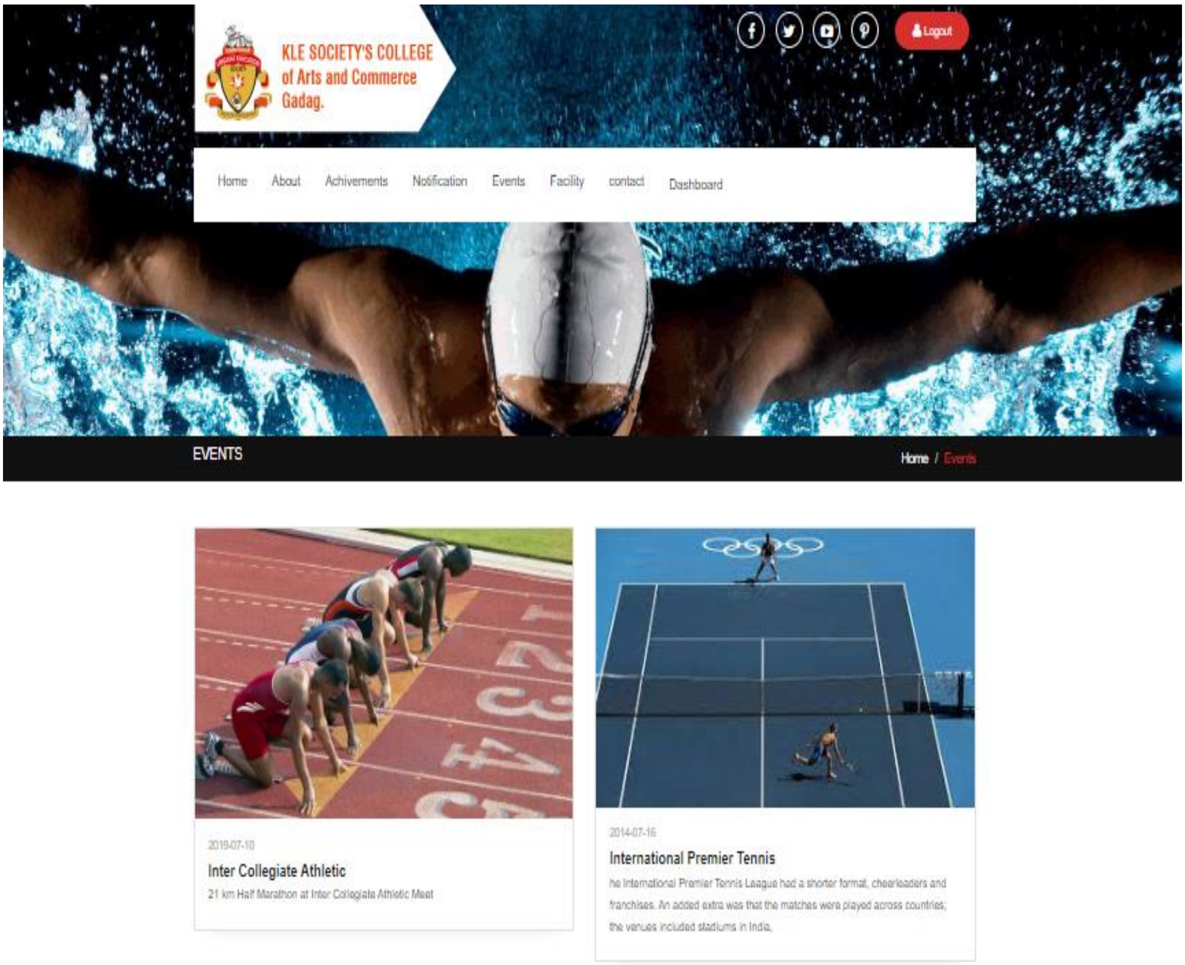
Mr. Snadeep Navalekar of B.com V sem secured 2nd place in 1500mtr run secured 2nd place and 2nd place in 21 km Half Marathon at Inter Collegiate Athletic Meet

half marathon

2018-07-09

Ms. Meghana of B.A I sem secured I st place in 5000 km run & 21 km half marathon at Inter Collegiate Athletic Meet.

## EVENT PAGE:



2019-07-10

Inter Collegiate Athletic

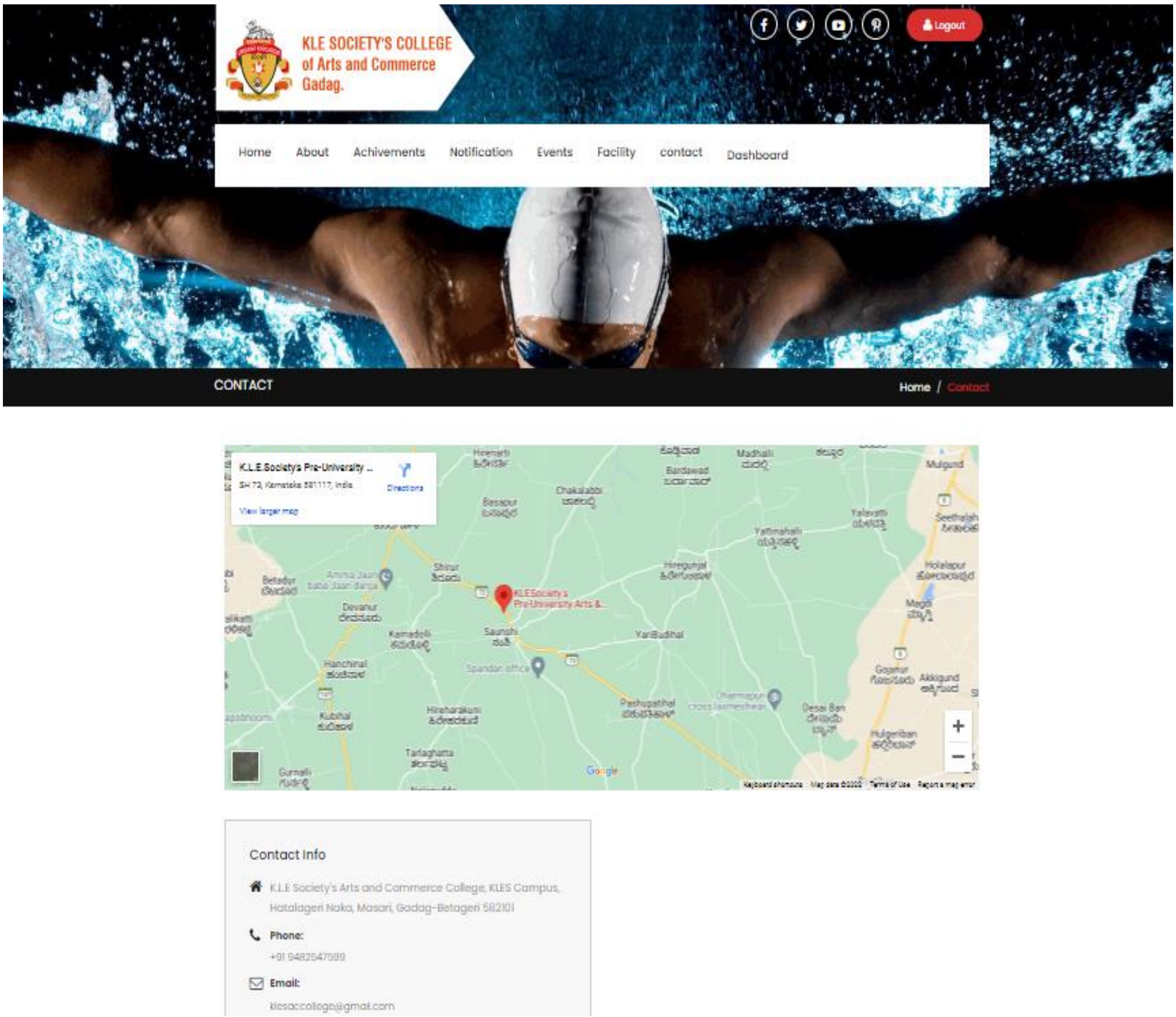
21 km Half Marathon at Inter Collegiate Athletic Meet

2014-07-16

International Premier Tennis

he International Premier Tennis League had a shorter format, cheerleaders and franchises. An added extra was that the matches were played across countries; the venues included stadiums in India.

## CONTACT PAGE:



Contact Info

K.L.E Society's Arts and Commerce College, KLES Campus, Hatalageri Naka, Masari, Gadag-Betageri 582101

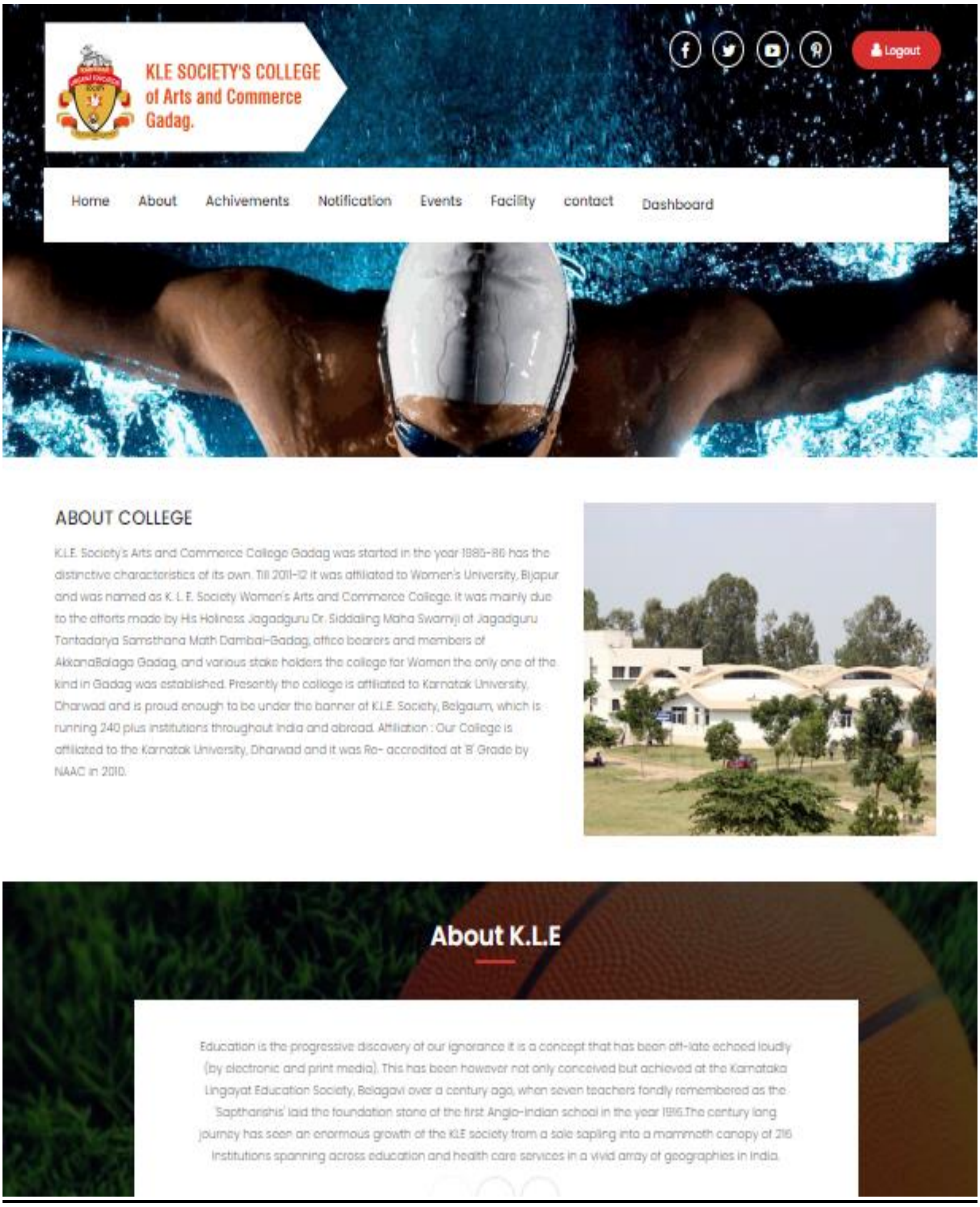
Phone:

+91 9482547599

Email:

klesaccollege@gmail.com

## ABOUT PAGE:



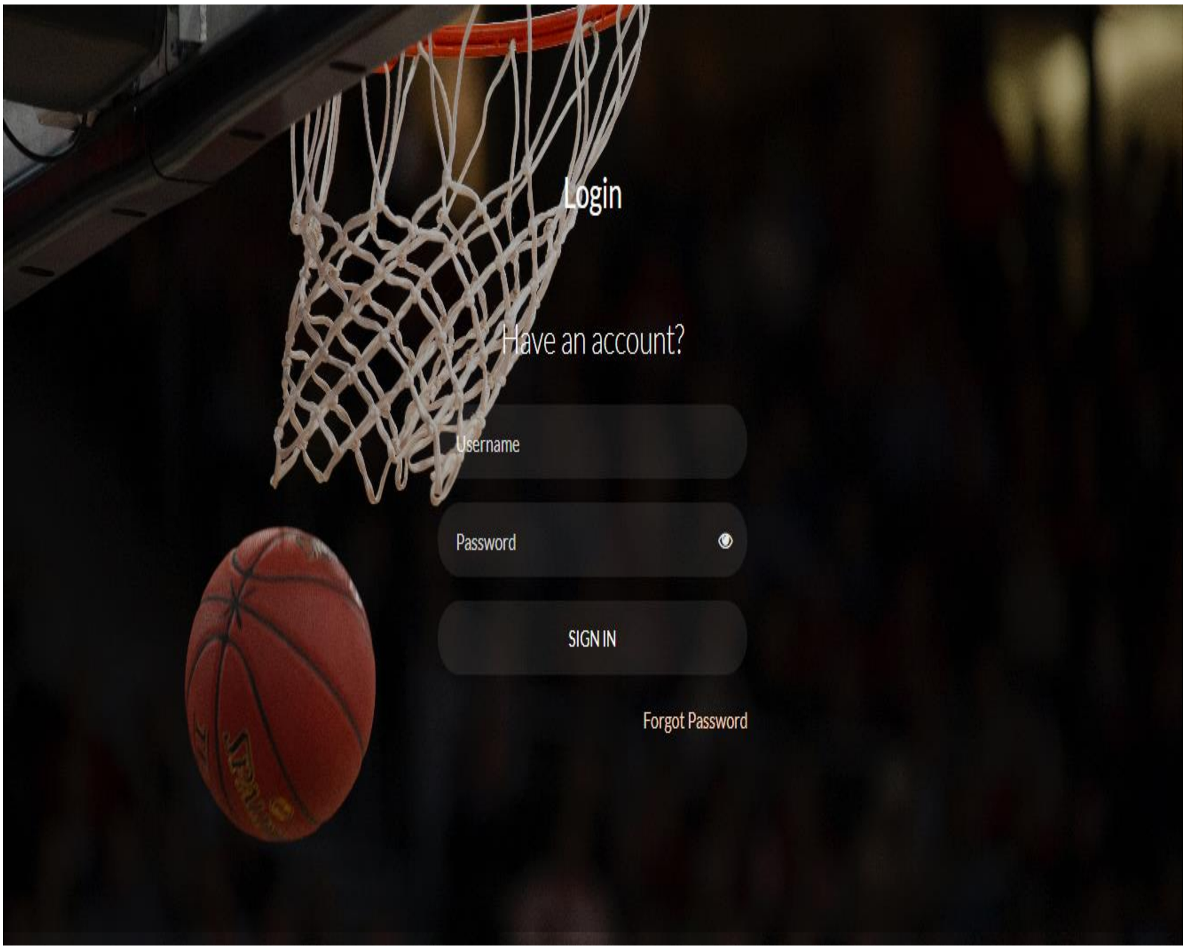
ABOUT COLLEGE

K.L.E. Society's Arts and Commerce College Gadag was started in the year 1985-86 has the distinctive characteristics of its own. Till 2011-12 it was affiliated to Women's University, Bijapur and was named as K. L. E. Society Women's Arts and Commerce College. It was mainly due to the efforts made by His Holiness Jagadguru Dr. Siddaling Maha Swamiji of Jagadguru Tontadarya Samsthana Math Dambal-Gadag, office bearers and members of AkkanaBalaga Gadag, and various stake holders the college for Women the only one of the kind in Gadag was established. Presently the college is affiliated to Karnatak University, Dharwad and is proud enough to be under the banner of K.L.E. Society, Belgaum, which is running 240 plus institutions throughout India and abroad. Affiliation : Our College is affiliated to the Karnatak University, Dharwad and it was Re- accredited at 'B' Grade by NAAC in 2010.

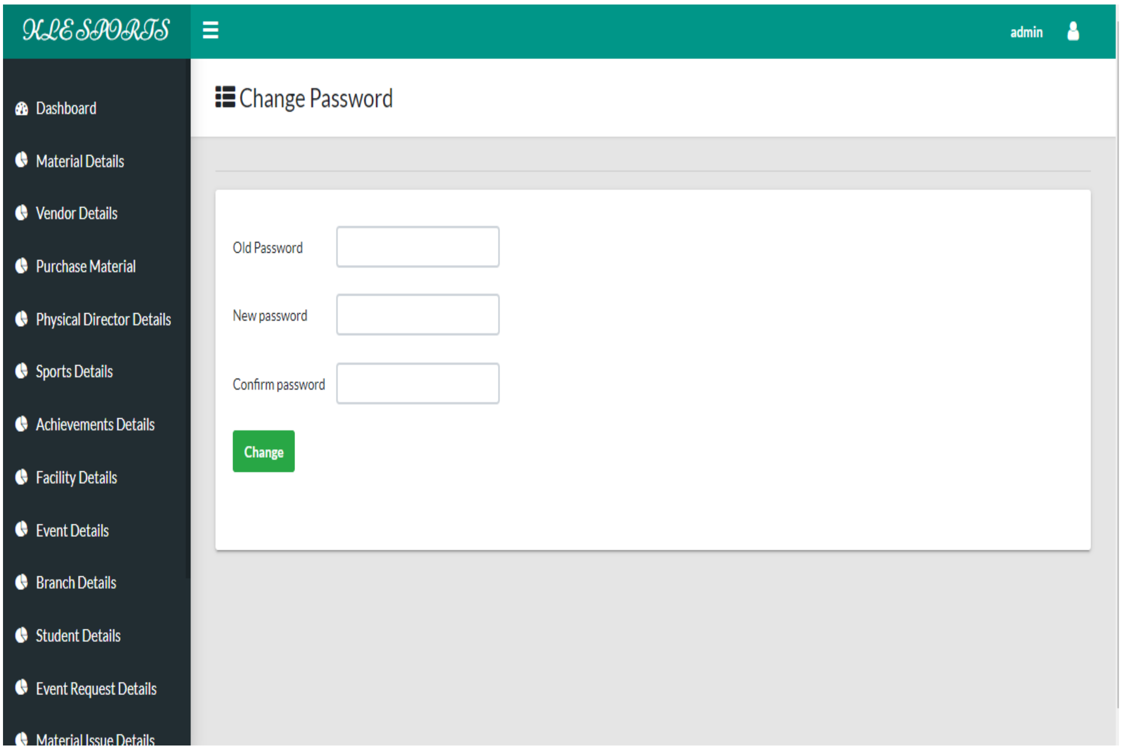
About K.L.E

Education is the progressive discovery of our ignorance it is a concept that has been off-late echoed loudly (by electronic and print media). This has been however not only conceived but achieved at the Karnataka Lingayat Education Society, Belagavi over a century ago, when seven teachers fondly remembered as the 'Saptharishis' laid the foundation stone of the first Anglo-Indian school in the year 1916.The century long journey has seen an enormous growth of the KLE society from a sole sapling into a mammoth canopy of 216 Institutions spanning across education and health care services in a vivid array of geographies in India.

## LOGIN PAGE:


Login
Have an account?
Username
Password
SIGN IN
Forgot Password

## CHANGE PASSWORD PAGE:



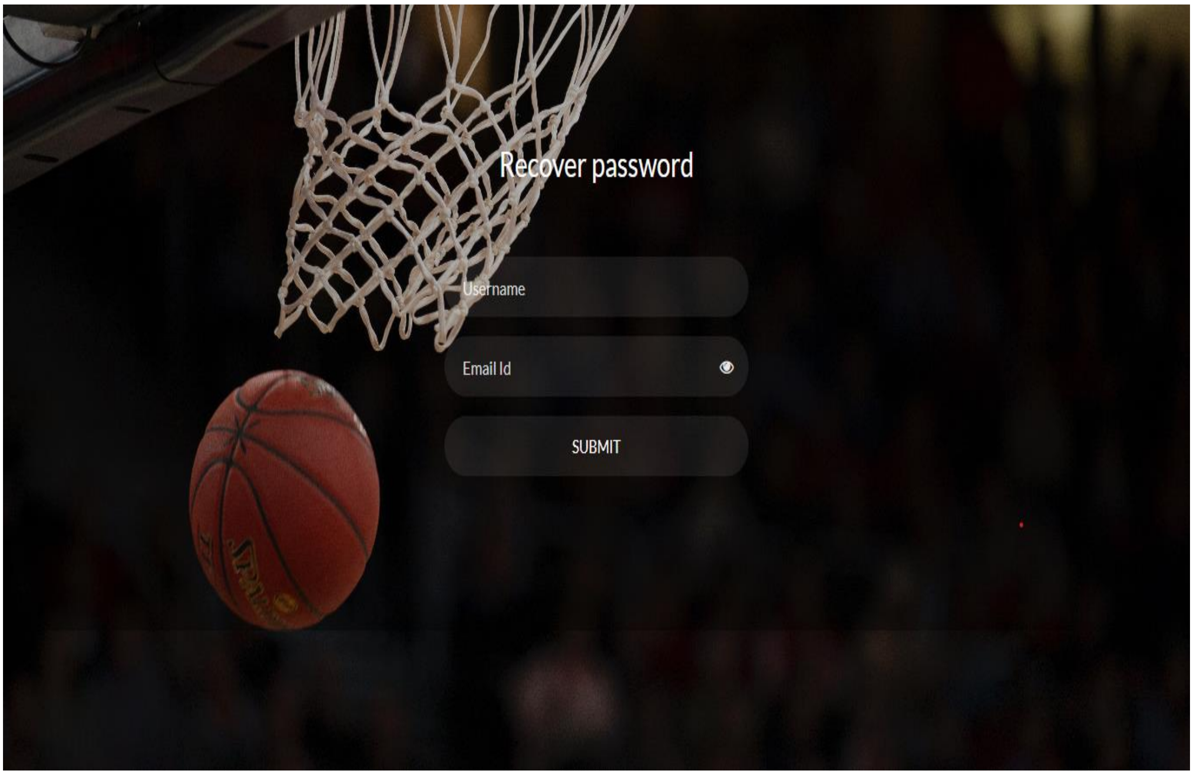
Change Password

Old Password

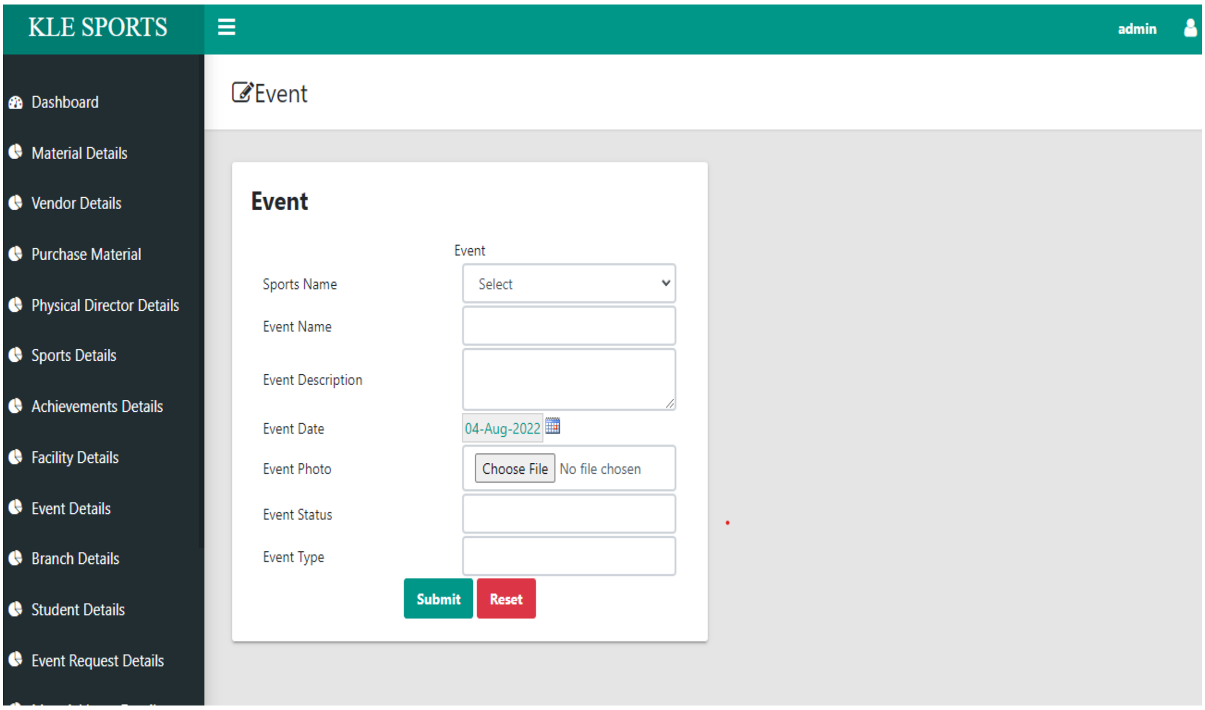
New password

Confirm password

## FORGOT PASSWORD  PAGE:



## ADD EVENTS PAGE:

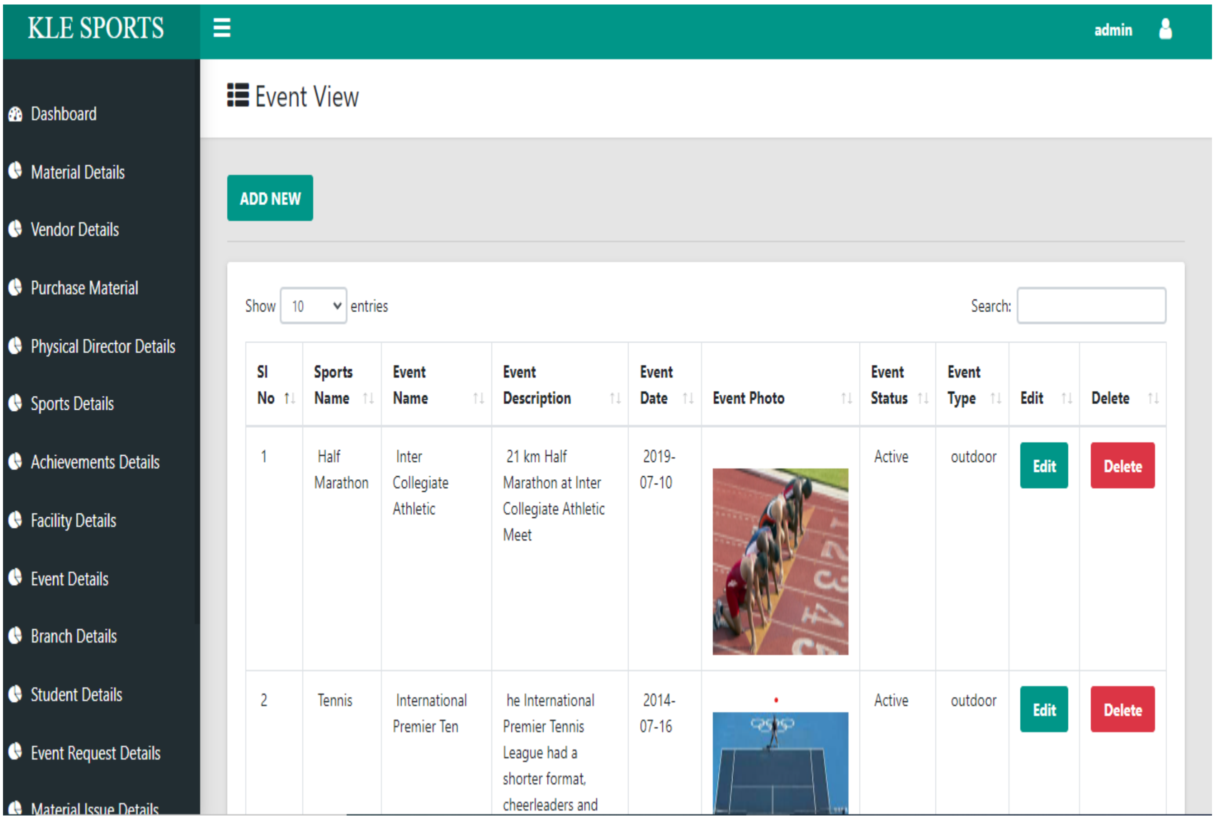


Event

Event

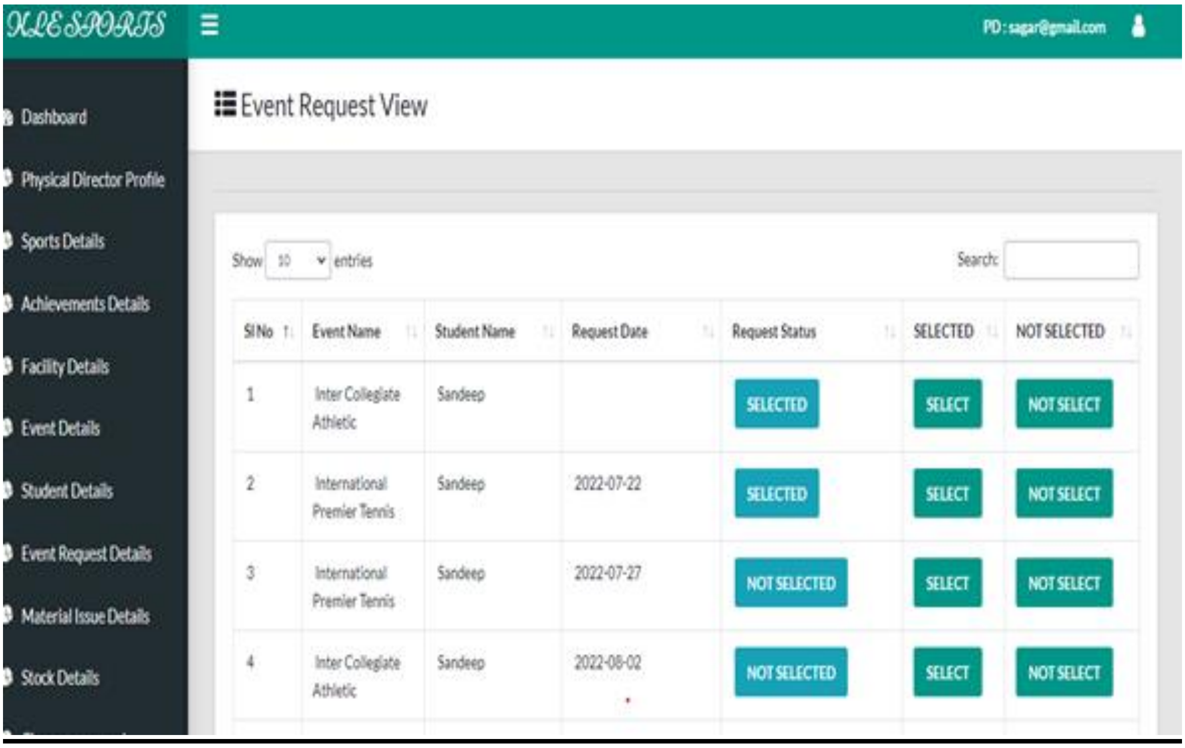
| | |
|---|---|
| Sports Name | Select |
| Event Name | |
| Event Description | |
| Event Date | 04-Aug-2022 |
| Event Photo | Choose File No file chosen |
| Event Status | |
| Event Type | |

## EVENTS VIEW PAGE:



| Sl No | Sports Name | Event Name | Event Description | Event Date | Event Photo | Event Status | Event Type | Edit | Delete |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Half Marathon | Inter Collegiate Athletic | 21 km Half Marathon at Inter Collegiate Athletic Meet | 2019-07-10 | | Active | outdoor | Edit | Delete |
| 2 | Tennis | International Premier Ten | he International Premier Tennis League had a shorter format, cheerleaders and | 2014-07-16 | | Active | outdoor | Edit | Delete |

## EVENT REQUEST:

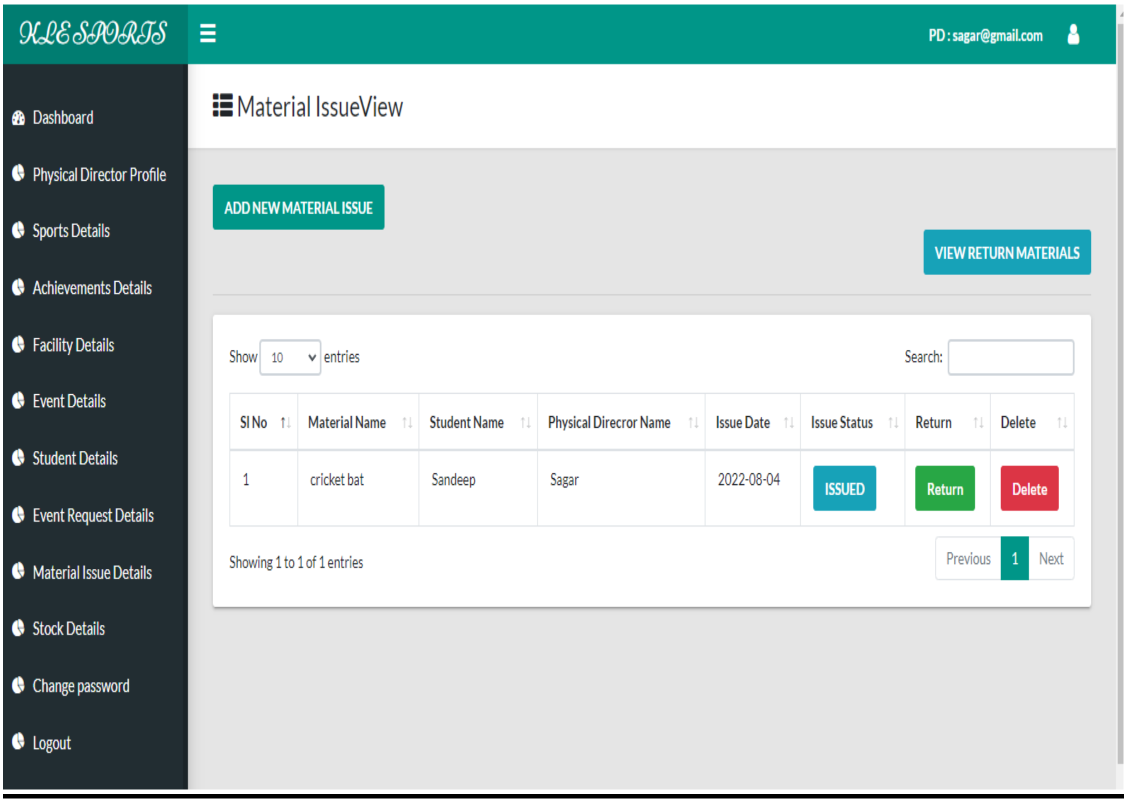


Event Request View

Show 10 entries Search:

| Sl No | Event Name | Student Name | Request Date | Request Status | SELECTED | NOT SELECTED |
|---|---|---|---|---|---|---|
| 1 | Inter Collegiate Athletic | Sandeep | | SELECTED | SELECT | NOT SELECT |
| 2 | International Premier Tennis | Sandeep | 2022-07-22 | SELECTED | SELECT | NOT SELECT |
| 3 | International Premier Tennis | Sandeep | 2022-07-27 | NOT SELECTED | SELECT | NOT SELECT |
| 4 | Inter Collegiate Athletic | Sandeep | 2022-08-02 | NOT SELECTED | SELECT | NOT SELECT |

# MATERIALS ISSUE:

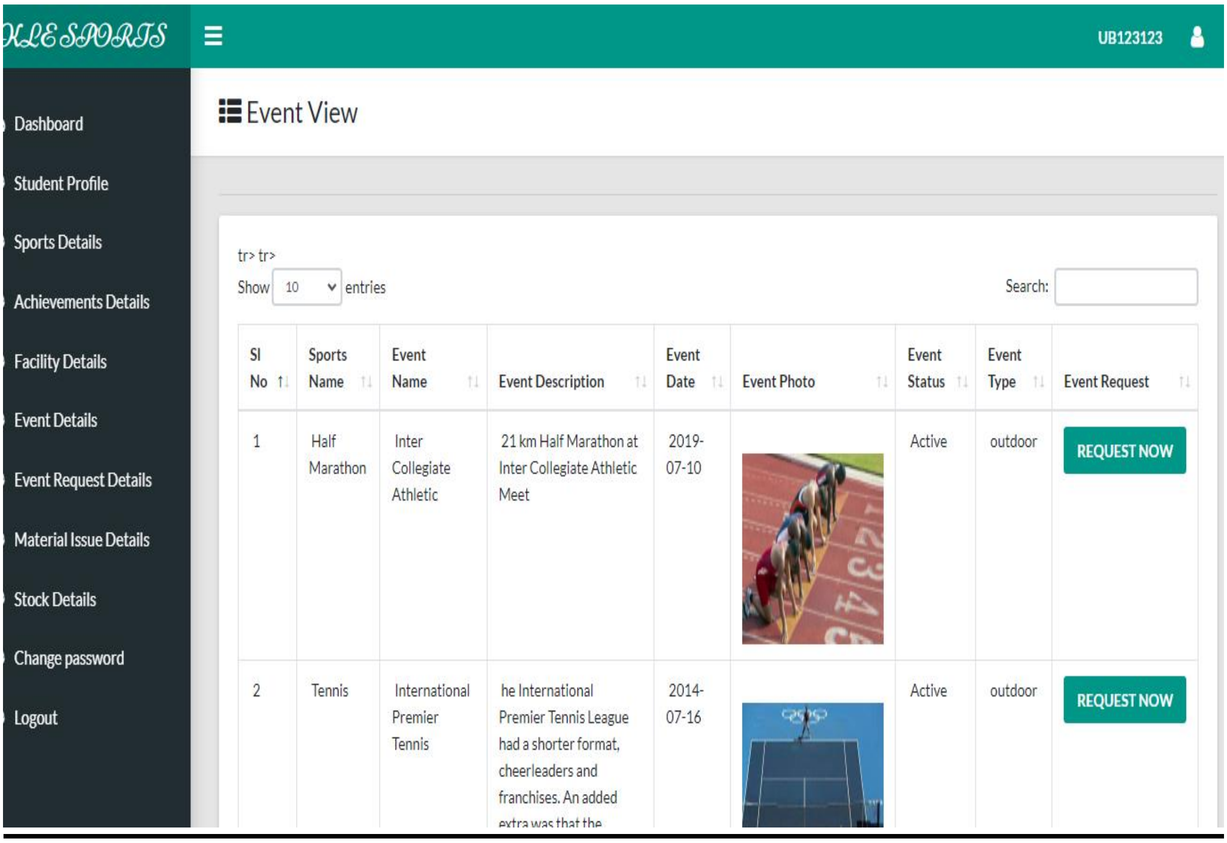


Material IssueView

| Sl No | Material Name | Student Name | Physical Direcror Name | Issue Date | Issue Status | Return | Delete |
|---|---|---|---|---|---|---|---|
| 1 | cricket bat | Sandeep | Sagar | 2022-08-04 | ISSUED | Return | Delete |

Showing 1 to 1 of 1 entries

## EVENT VIEW FOR STUDENT:



Event View

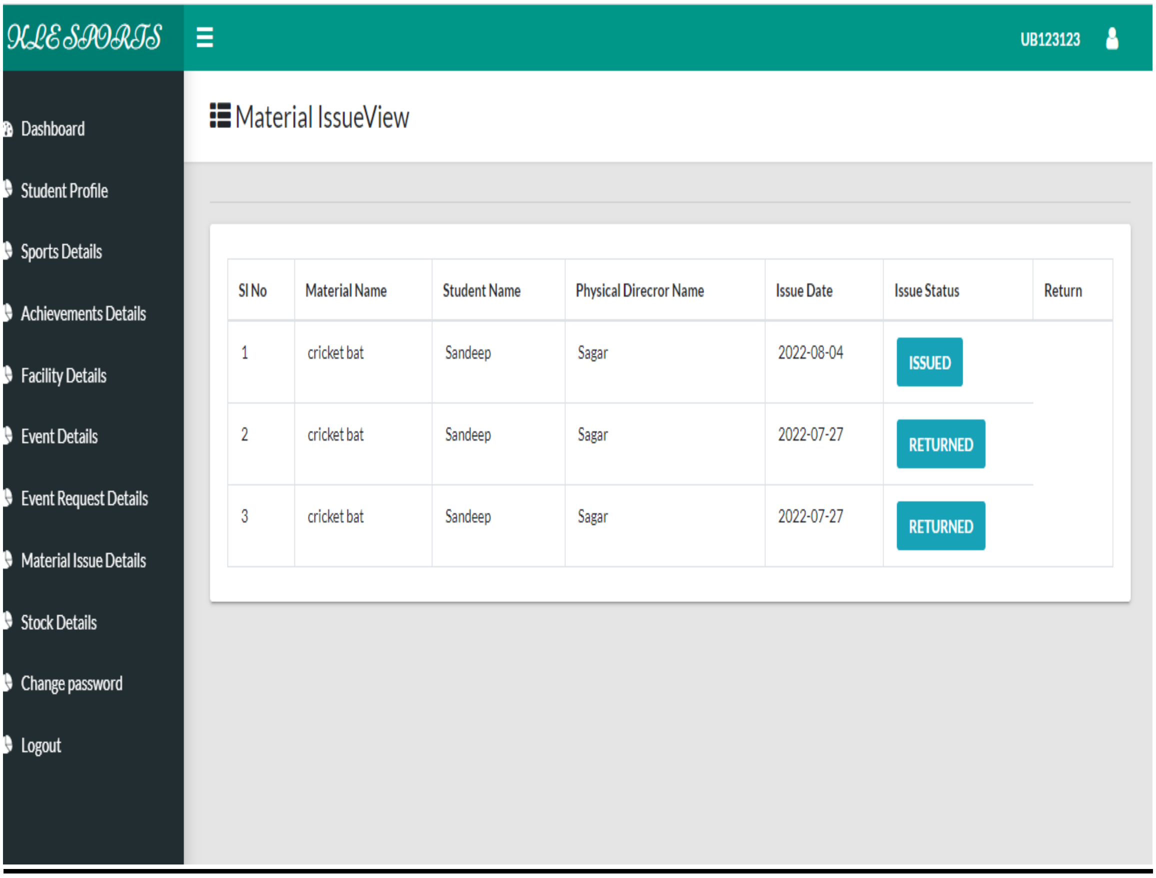
tr> tr>

Show 10 entries

Search:

| Sl No | Sports Name | Event Name | Event Description | Event Date | Event Photo | Event Status | Event Type | Event Request |
|---|---|---|---|---|---|---|---|---|
| 1 | Half Marathon | Inter Collegiate Athletic | 21 km Half Marathon at Inter Collegiate Athletic Meet | 2019-07-10 | | Active | outdoor | REQUEST NOW |
| 2 | Tennis | International Premier Tennis | he International Premier Tennis League had a shorter format, cheerleaders and franchises. An added extra was that the | 2014-07-16 | | Active | outdoor | REQUEST NOW |

## MATERIAL ISSUE VIEW FOR STUDENT:



Material IssueView

| Sl No | Material Name | Student Name | Physical Direcror Name | Issue Date | Issue Status | Return |
|---|---|---|---|---|---|---|
| 1 | cricket bat | Sandeep | Sagar | 2022-08-04 | ISSUED | |
| 2 | cricket bat | Sandeep | Sagar | 2022-07-27 | RETURNED | |
| 3 | cricket bat | Sandeep | Sagar | 2022-07-27 | RETURNED | |

# SOURCE CODE

## DATABASE CONNECTION:

```
<?php

$server='localhost';

$u_name='root';

$pass=";

$db='sports_management';

$conn=new mysqli($server,$u_name,$pass,$db);

?>
```

## LOGIN AUTHENTICATION:

```
<?php session_start();?>

<?php



include("db_connect/db_connect.php");

$username=$_REQUEST['username'];

$password=$_REQUEST['password'];
```

```
$dat=date('Y-m-d h:m:r');

$sql="select * from login where username='$username' and password='$password'";

$res=mysqli_query($conn,$sql);

if($row=mysqli_fetch_array($res))

{

$type=$row['type'];

$_SESSION['uname']=$username;

$_SESSION['type']=$type;

if($type=="admin")

{

header('location:admin/home.php');

}

else if($type=="student")

{

header('location:student/home.php');

}

else if($type=="pd")
{

header('location:pd/home.php');

}

}

else

{

?>

<script>

alert("Invalid Username Or Password");

history.back();

</script>

<?php

}

?>
```

## CHANGE PASSWORD:

```
<?php



session_start();

?>
```

```
<?php

  $a=$_POST['new_pwd'];

  $b=$_POST['conf_pwd'];

   include('../db_connect/db_connect.php');

   $sql = "select * from  login where username ='".$_SESSION['uname']."'

     and    password='".$_POST['old_pwd']."'";

    $res=mysqli_query($conn,$sql);

if($a==$b)

{

if(mysqli_num_rows($res)>0)

{

$sql1 = "update login set password = '".$_POST['new_pwd']."' where username
='".$_SESSION['uname']."'";

$res1 = mysqli_query($conn,$sql1);

?>

<script>

alert('Successfully updated your password');

document.location="change pwd.php";

</script>
<?php

} else

{

?>

<script>

alert('Failed to update your password');

document.location="change pwd.php";

</script>

<?php

}

}else

{

?>

<script>

alert('New Password and Retype Password do not match');

document.location="change pwd.php";

</script>

<?php
```

```
}

?>
```

## FORM:

```
<!DOCTYPE html>

<html lang="en">

<style type="text/css">

<!--

.style3 {font-size: medium; font-style: italic; font-weight: bold; }

-->

</style>

<?php include('meta_tag.php');?>

</head>

<body class="app sidebar-mini rtl">

<!-- Navbar-->

<?php include('header.php');?>

<!-- Sidebar menu-->

<?php include('sidebar.php'); ?>

<main class="app-content">
```

```
<div class="app-title">

<div>

<h1><i class="fa fa-edit"></i>Student</h1>

</div>

</div>

<div class="row">

<div class="col-md-12">

<div class="tile">

<h3 class="tile-title">Student</h3>

<div class="tile-body">

<?php include("cal.php"); ?>

<?php

include('val.php');

?>

<form action="student_insert.php" enctype="multipart/form-data" id="formID"
method="post" name="form1">

<table width="450" height="919" border="0" align="center">

<tr>

<td colspan="2"><div align="center"><strong>Student</strong></div></td>
```

```
</tr>

<tr>

<td width="174"><span class="draggable sticky">Student Name</span></td>

<td width="166"><input name="student_full_name"
class="validate[required,custom[onlyLetter]] form-control" type="text"
id="student_full_name"></td>

</tr>

<tr>

<td><span class="draggable sticky">USN No</span></td>

<td><input name="usn_no"
class="validate[required,custom[noSpecialCaracters]] form-control" type="text"
id="usn_no"></td>

</tr>

<tr>

<td><span class="draggable sticky">Address</span></td>

<td><textarea name="address" class="validate[required] form-control"
id="address"></textarea></td>

</tr>

<tr>

<td><span class="draggable sticky">Email Id</span></td>
```

```
<td><input name="email_id" class="validate[required,custom[email]] form-control" type="email" id="email_id"></td>

</tr>

<tr>

<td><span class="draggable sticky text-right">Contact No</span></td>

<td><input name="contact_no" class="validate[required,custom[mobile]] form-control" type="text" id="contact_no"></td>

</tr>

<tr>

<td><span class="draggable sticky">Gender</span></td>

<td><input name="gender" type="radio" value="Male">

Male

<input name="gender"  type="radio" value="Female">

Female</td>

</tr>

<tr>

<td><span class="draggable sticky">Date Of Birth</span></td>

<td><?php

$date_default = "";
```

```
if(isset($row['date']))

{

$date_default =$row['date'];

}

else

{

$date_default =date('Y-m-d');

}

$myCalendar = new tc_calendar("date_of_birth", true, false);

$myCalendar->setIcon("calendar/images/iconCalendar.gif");

$myCalendar->setDate(date('d', strtotime($date_default))

, date('m', strtotime($date_default))

, date('Y', strtotime($date_default)));

$myCalendar->setPath("calendar/");

$myCalendar->setYearInterval(1800, date('Y'));

$myCalendar->setAlignment('left', 'bottom');

$myCalendar->writeScript();
```

```
?> </td>
</tr>
<tr>
<td><span class="draggable sticky">Admission Year</span></td>
<td>
<?php
$date_default = "";
if(isset($row['date']))
{
$date_default =$row['date'];
}
else
{
$date_default =date('Y-m-d');
}
$myCalendar = new tc_calendar("admission_year", true, false);
$myCalendar->setIcon("calendar/images/iconCalendar.gif");
$myCalendar->setDate(date('d', strtotime($date_default))
, date('m', strtotime($date_default))

, date('Y', strtotime($date_default)));

$myCalendar->setPath("calendar/");

$myCalendar->setYearInterval(1800, date('Y'));

$myCalendar->setAlignment('left', 'bottom');

$myCalendar->writeScript();

?> </td>

</tr>

<tr>

<td><span class="draggable sticky">Branch Name </span></td>

<td><select name="branch_id" class="validate[required] form-control"
id="branch_id">

<option value="">select</option>

<?php

include('../db_connect/db_connect.php');

$sql="select * from branch";

$res=mysqli_query($conn,$sql);

while($row=mysqli_fetch_array($res))

{
```

```
?>

<option value="<?php echo $row['branch_id'];?>"><?php echo
$row['branch_name'];?></option>

<?php

}

?>

</select></td>

</tr>

<tr>

<td><span class="draggable sticky">Student Photo</span></td>

<td><input class="validate[required] form-control" name="student_photo"
type="file" id="student_photo"></td>

</tr>

<tr>

<td colspan="2"><div align="center">

<input type="submit" class="btn btn-primary" name="Submit" value="Submit">

<input class="btn btn-danger" type="reset" name="Reset" value="Reset">

</div></td>

</tr>
```

```
</table>

</form>

</div>

</div>

</div>

</main>

<?php include('footer.php');?>

<!-- Essential javascripts for application to work-->

<?php include('script.php');?>

<?php include('val.php');?>

</body>

</html>
```

## INSERT:

```
<?php

include('../db_connect/db_connect.php');

$student_full_name=$_POST['student_full_name'];

$usn_no=$_POST['usn_no'];

$address=$_POST['address'];

$email_id=$_POST['email_id'];
```

```
$contact_no=$_POST['contact_no'];

$gender=$_POST['gender'];

$date_of_birth=$_POST['date_of_birth'];

$admission_year=$_POST['admission_year'];

$branch_id=$_POST['branch_id'];

$student_photo=$_POST['student_photo'];

$student_photo=$_FILES['student_photo']['name'];

$tmp_location=$_FILES['student_photo']['tmp_name'];

$target="../images/".$student_photo;

move_uploaded_file($tmp_location,$target);

$sql="insert into student
values(null,'$student_full_name','$usn_no','$address','$email_id','$contact_no','$ge
nder','$date_of_birth','$admission_year','$branch_id','$student_photo')";

mysqli_query($conn,$sql);

$sql2="insert into login values(null,'$usn_no','$usn_no','student','Enter Your email
Id','$email_id','Active')";

mysqli_query($conn,$sql2);

?>

<script>

alert("Student Is Added..");
```

```
document.location="student_view.php";

</script>
```

## VIEW:

```
<!DOCTYPE html>

<html lang="en">

<head>

<?php include('meta_tag.php');?>

</head>

<body class="app sidebar-mini rtl">

<!-- Navbar-->

<?php include('header.php');?>

<!-- Sidebar menu-->

<?php include('sidebar.php');?>

<main class="app-content">

<div class="app-title">

<div>

<h1><i class="fa fa-th-list"></i> <span class="draggable sticky">Student</span>
View</h1>

<!-- <p>Table to display analytical data effectively</p> -->
```

```
<p></p>

</div>

<!--  <ul class="app-breadcrumb breadcrumb side">

<li class="breadcrumb-item"><i class="fa fa-home fa-lg"></i></li>

<li class="breadcrumb-item">Tables</li>

<li class="breadcrumb-item active"><a href="#">Data Table</a></li>

</ul> -->

</div>

<a href="student_form.php" class="btn btn-primary">ADD NEW</a>

<hr>

<div class="row">

<div class="col-md-14">

<div class="tile">

<div class="tile-body">

<table class="table table-hover table-bordered" id="sampleTable">

<thead>

<tr>

<th>Sl No </th>
```

```
<th><span class="draggable sticky">Student Name</span></th>

<th><span class="draggable sticky">USN No</span></th>

<th><span class="draggable sticky">Address</span></th>

<th><span class="draggable sticky">Email Id</span></th>

<th><span class="draggable sticky right">Contact No</span></th>

<th><span class="draggable sticky">Gender</span></th>

<th><span class="draggable sticky">Date Of Birth</span></th>

<th><span class="draggable sticky">Admission Year</span></th>

<th><span class="draggable sticky">Branch Name</span></th>

<th><span class="draggable sticky">Student Photo</span></th>

<th>Edit</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<?php

include('../db_connect/db_connect.php');

$sql="select * from student sd,branch br where sd.branch_id=br.branch_id";
```

```
$res=mysqli_query($conn,$sql);

$no=1;

while($row=mysqli_fetch_array($res))

{

?>

<tr>

<td>&nbsp;<?php echo $no++;?></td>

<td>&nbsp;<?php echo $row['student_full_name'];?></td>

<td>&nbsp;<?php echo $row['usn_no'];?></td>

<td>&nbsp;<?php echo $row['address'];?></td>

<td>&nbsp;<?php echo $row['email_id'];?></td>

<td>&nbsp;<?php echo $row['contact_no'];?></td>

<td>&nbsp;<?php echo $row['gender'];?></td>

<td>&nbsp;<?php echo $row['date_of_birth'];?></td>

<td>&nbsp;<?php echo $row['admission_year'];?></td>

<td>&nbsp;<?php echo $row['branch_name'];?></td>

<td>&nbsp;<img src="../images/<?php echo $row['student_photo'];?>"
width="100" height="120"></td>
<td><a href="student_edit.php?student_id=<?php echo
$row['student_id'];?>">Edit</a></td>

<td><a href="student_delete.php?student_id=<?php echo
$row['student_id'];?>">Delete</a></td>

</tr>

<?php

}

?>

</tbody>

</table>

</div>

</div>

</div>

</div>

  </main>

<!-- Essential javascripts for application to work-->

<?php include('script.php');?>

<?php include('footer.php'); ?>

</body>
```

```
</html>
```

## EDIT:

```
<!DOCTYPE html>
<html lang="en">
<style type="text/css">
<!--
.style3 {font-size: medium; font-style: italic; font-weight: bold; }
-->
</style>
<?php include('meta_tag.php');?>
</head>
<body class="app sidebar-mini rtl">
<!-- Navbar-->
<?php include('header.php');?>
<!-- Sidebar menu-->
<?php include('sidebar.php'); ?>
<main class="app-content">
<div class="app-title">
<div>
<h1><i class="fa fa-edit"></i><strong>Student</strong></h1>
</div>
</div>
<div class="row">
<div class="col-md-6">
<div class="tile">
```

```
<h3 class="tile-title"><strong>Student</strong></h3>

<div class="tile-body">

<?php include("cal.php"); ?>

<?php

include('val.php');

?>

<?php

include('../db_connect/db_connect.php');

$student_id=$_REQUEST['student_id'];

$sql="select * from student where student_id=$student_id";

$res=mysqli_query($conn,$sql);

$row=mysqli_fetch_array($res);

?>

<form id="formID" enctype="multipart/form-data" action="student_update.php"
method="post" name="form1">

<input type="hidden" name="student_id" value="<?php echo
$row['student_id'];?>">

<table width="450" border="0" align="center">

<tr>

<td colspan="2"><div align="center"><strong>Student</strong></div></td>

</tr>

<tr>

<td width="174"><span class="draggable sticky">Student Full
Name</span></td>

<td width="166"><input name="student_full_name"
class="validate[required,custom[onlyLetter]] form-control" type="text"
id="student_full_name" value="<?php echo $row['student_full_name'];?>"></td>
```

```
</tr>

<tr>

<td><span class="draggable sticky">USN No</span></td>

<td><input name="usn_no" class="validate[required,custom[noSpecialCaracters]]
form-control" type="text" id="usn_no" value="<?php echo
$row['usn_no'];?>"></td>

</tr>

<tr>

<td><span class="draggable sticky">Address</span></td>

<td><textarea name="address" class="validate[required] form-control"
id="address"><?php echo $row['address'];?></textarea></td>

</tr>

<tr>

<td><span class="draggable sticky">Email Id</span></td>

<td><input name="email_id" class="validate[required,custom[email]] form-
control" type="email" id="email_id" value="<?php echo
$row['email_id'];?>"></td>

  </tr>

  <tr>

    <td><span class="draggable sticky text-right">Contact No</span></td>

    <td><input name="contact_no" class="validate[required,custom[mobile]]
form-control" type="text" id="contact_no" value="<?php echo
$row['contact_no'];?>"></td>

  </tr>

  <tr>

<td><span class="draggable sticky">Gender</span></td>

<td><input name="gender" type="radio" value="Male">

Male
```

```
<input name="gender" type="radio" value="Female">

Female</td>

</tr>

<tr>

<td><span class="draggable sticky">Date Of Birth</span></td>

<td>

<?php

$date_default = "";


if(isset($row['date_of_birth']))

{

$date_default =$row['date_of_birth'];

}

else

{

$date_default =date('Y-m-d');

}

$myCalendar = new tc_calendar("date_of_birth", true, false);

$myCalendar->setIcon("calendar/images/iconCalendar.gif");

$myCalendar->setDate(date('d', strtotime($date_default))

, date('m', strtotime($date_default))

, date('Y', strtotime($date_default)));


$myCalendar->setPath("calendar/");

$myCalendar->setYearInterval(1800, date('Y'));

$myCalendar->setAlignment('left', 'bottom');
```

```
$myCalendar->writeScript();

?></td>

</tr>

<tr>

<td><p class="draggable sticky">Admission Year</p>

</td>

<td>

<?php

$date_default = "";

if(isset($row['admission_year']))

{

$date_default =$row['admission_year'];

}

else

{

$date_default =date('Y-m-d');

}

$myCalendar = new tc_calendar("admission_year", true, false);

$myCalendar->setIcon("calendar/images/iconCalendar.gif");

$myCalendar->setDate(date('d', strtotime($date_default))

, date('m', strtotime($date_default))

, date('Y', strtotime($date_default)));

$myCalendar->setPath("calendar/");

$myCalendar->setYearInterval(1800, date('Y'));

$myCalendar->setAlignment('left', 'bottom');

$myCalendar->writeScript();
```

```
?></td>

</tr>

<tr>

<td><span class="draggable sticky">Branch Name </span></td>

<td>

<select name="branch_id" class="validate[required] form-control"
id="branch_id">

<option value="">Branch Name</option>

<?php

include('../db_connect/db_connect.php');

$sql="select * from branch";

$res=mysqli_query($conn,$sql);

while($row2=mysqli_fetch_array($res))

{

?>

<option value="<?php echo $row2['branch_id'];?>" <?php
if($row['branch_id']==$row2['branch_id']){ echo "selected";} ?>><?php echo
$row2['branch_name'];?></option>

<?php

}

?>

</select></td>

</tr>

<tr>

<td><span class="draggable sticky">Student Photo</span></td>

<td><input name="student_photo" class="validate[required] form-control"
type="file" id="student_photo"></td>
```

```
</tr>
<tr>
<td colspan="2"><div align="center">
<input type="submit" class="btn btn-primary" name="Submit" value="Submit">
<input type="reset" class="btn btn-danger" name="Reset" value="Reset">
</div></td>
</tr>
</table>
</form>
</div>
</div>
</div>
</main>
<?php include('footer.php');?>
<!-- Essential javascripts for application to work-->
<?php include('script.php');?>
<?php include('val.php');?>
</body>
</html>
```

## DELETE:

```
<?php

include('../db_connect/db_connect.php');

$student_id=$_REQUEST['student_id'];

$sql="delete from student where student_id=$student_id";
```

```
mysqli_query($conn,$sql);

?>

<script>

alert('values are deleted');

document.location='student_view.php';

</script>
```

## UPDATE:

```
<?php

include('../db_connect/db_connect.php');

$student_id=$_POST['student_id'];

$student_full_name=$_POST['student_full_name'];

$usn_no=$_POST['usn_no'];

$address=$_POST['address'];

$email_id=$_POST['email_id'];

$contact_no=$_POST['contact_no'];

$gender=$_POST['gender'];

$date_of_birth=$_POST['date_of_birth'];

$admission_year=$_POST['admission_year'];

$branch_id=$_POST['branch_id'];
```

```
$student_photo=$_FILES['student_photo']['name'];

$tmp_location=$_FILES['student_photo']['tmp_name'];

$target="../images/".$student_photo;

move_uploaded_file($tmp_location,$target);

$sql="update student set
student_full_name='$student_full_name',usn_no='$usn_no',address='$address',em
ail_id='$email_id',contact_no='$contact_no',gender='$gender',date_of_birth='$dat
e_of_birth',admission_year='$admission_year',branch_id='$branch_id',student_ph
oto='$student_photo' where student_id='$student_id'";

mysqli_query($conn,$sql);

?>

<script>

alert("values updated..");

document.location="student_view.php";

</script>
```

# TESTING

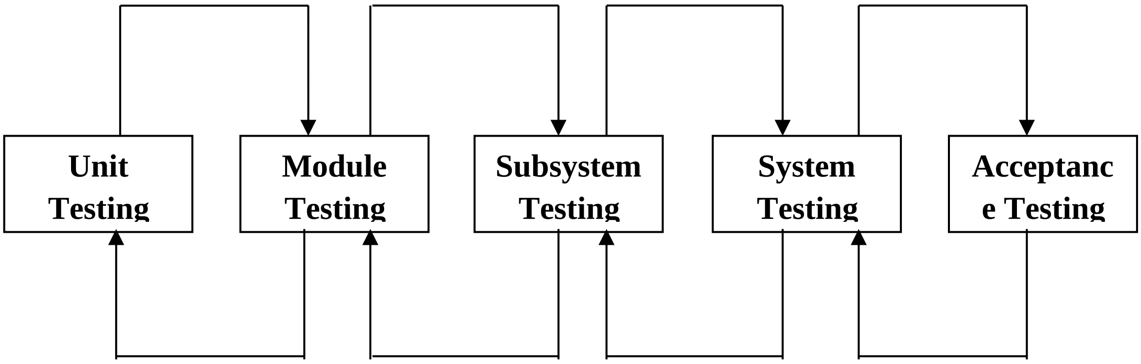
## Introduction:

Software Testing is a process of executing program with an indent of finding error. Testing is vital to success of the system. Testing demonstrates that the software functions appear to be working according to the specifications and performance requirements appeared to have been met. If a test is conducted successfully, it will discover errors in the software.

Software Testing consists of all test life cycle activities like static and dynamic testing concerned with planning, preparation and evaluation of software products to determine that the software products satisfy customers requirements and are fit the customer use.

The various strategies that were used in testing this software are as follows:

- Unit Testing
- Integration Testing
- System Testing
- Validation Testing
- Black Box Testing
- White Box Testing
- ➢ Acceptance Testing



## TYPES OF TESTS

### ➢ UNIT TESTING:

Unit testing is done on individual modules as they are completed and become executable. This system was tested with the set of proper test data for each module and results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module.

This is also known as MODULE TESTING. This testing is carried out during phases, each module is founded to be working satisfactory as regards to the expected result from the module.

Unit testing involves the design of the test cases that validate the internal program logic is functioning properly, and that program input produces valid output. All decision branches and internal code flow should be validated.

### ➢ INTEGRATION TESTING:

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied as shown by the Unit testing, the combination of the components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.Integration Testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests to aggregates and delivers as its output. The Integration Testing verifies functional, performance, and reliablility requirements placed on a major design items.

### ➢ FUNCTIONAL TESTING:

Functional tests provide a systematic demonstration of the functions tested that are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional Testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted.

Invalid Input : Identified classes of invalid input must be accepted.

Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised.

Systems/Procedures : Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of the current test is determined.

## ➢ SYSTEM TESTING:

In this the entire software system is tested. The reference document for this process is the requirement document. Here the entire software is tested and the performance of the system was observed to see that it satisfies the requirement specification.

System testing tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

System testing involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user. The system meets all requirements of the client’s specifications.The following are the types of system tests that were carried out for the system:

### • Validation Testing:

The system has been tested and implemented successfully and thus ensured that all requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

### • Black Box Testing:

This method focuses on the functional requirements of the software. This testing enables to derive set input conditions that will fully exercise all functional requirements of the program. Black Box Testing attempts to find errors in the following category.

- Incorrect or missing functions.
- Interface errors.
- Error in external database access.
- Performance errors.
- Initialization and Termination errors.

## White Box Testing:

This is performed early in the testing process, while Black Box testing is applied during the last stage of testing. In this test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been used to generate the test case in the following test cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions from their True and False side.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.
- Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

## Performance Testing:

Performance Testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or software contributes most to the poor performance.

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

## ➢ Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors. It deals with successful satisfaction of user needs. This project is approved and accepted by the clients. The process flow and execution is 99% working with respect to system testing procedure.

### Test Objectives:

- The system is tested with variety of inputs. The System is tested for accuracy and correctness of the results obtained. Finally the system is tested for inter-operability.
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

### Features to be tested

- Student Registration
- Admin Login
- Material Management
- Posting Events
- **Performance Testing** checks to see if the system runs within specified temporal and spatial constraints

**Regression Testing** checks to see if the latest fixes and enhancements broke something that used to work

**Stress Testing** sees how the system holds up under excessive load, or when run in an environment without reasonable resource

## Test Cases

### Login Module and Registration module

| Test Case No | Scenario | Expected output | Actual output | Pass/ Fail |
|---|---|---|---|---|
| TC-1 | Leave the username and password textbox | Should not login | (*)User<br>and password<br>field is empty | Pass |
| TC-2 | Enter invalid<br>password and<br>click login | Should not login | Password is<br>incorrect | Pass |
| TC-3 | Enter valid<br>password and<br>click login<br>button | Should not login | (*)User<br>field is empty | Pass |
| TC-4 | Password field<br>kept blank | Should not login | (*)Enter valid<br>Password | Pass |
| TC-5 | Email invalid | Should not register | Enter a valid email | Pass |
| TC-6 | Numbers less than 10 digits | Should not register | Enter a valid contact number | Pass |
| TC-7 | Password field left blank | Should not register | Enter password | Pass |

## Material

| Test Case No | Scenario | Expected output | Actual output | Pass/Fail |
|---|---|---|---|---|
| TC-8 | Material | Show material status | List of materials | Pass |
| TC-9 | material issue | Should show the material<br>That is issued to student | shows<br>The issue details | Pass |

## Student

|  |  |  |  |  |
|---|---|---|---|---|
| TC-10 | Event details | Show event details | List of events | Pass |
| TC-11 | Student details | Show Details | Details of student | Pass |

### Event

| Test Case No | Scenario | Expected output | Actual output | Pass /Fail |
|---|---|---|---|---|
| TC-12 | Event | View Events List | Should display all events | Pass |
| TC-13 | Event | Approve | Should approve from student | Pass |

# Conclusion

- Material management helps a lot store materials details about what is the quantity of materials left and what is needed.

- It is possible for the students to book a needed material online and reserve for them. Also they can request to conduct for the events

- However the admin needs to verify material allocation to student by referring to the ID card of students.

- The"K.L.E Sports Mangagement" helps to the department carry out all tasks online and saves the time.

- This is a college level project and needs to use to our college premises only it cannot be used by other college.

- The time to go to principal and other department for students and PD Head is saved through this website.

# BIBLIOGRAPHY

## BIBLIOGRAPHYAND REFERENCES:

- "Server side programming with PHP and MySQL- THE MISSING MANUAL SECOND EDITION By BREET McLaughlin.
- The "Software Engineering" Ivan Somerville 6th edition
- W3 CSS DEMO page which help in the designing of the forms.

Websites:

- http://www.google.com
- http://w3schools.com
- http://www.php.com