

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Jnana Sangama, Santhibastawad Road, Machhe**

**Belagavi - 590018, Karnataka, India**



## **DBMS LABORATORY WITH MINI PROJECT (18CSL58) REPORT ON**

### **“AICTE Activity Database Management”**

**Submitted in the partial fulfilment of the requirements for the award of the degree of**

## **BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING**

**For the Academic Year 2021-2022**

**Submitted by**

**ARUN M MIRLE  
ASHISH B R**

**1JS20IS024  
1JS20IS025**

**Under the Guidance of**

**Dr. Sowmya K N  
Associate Professor, Dept. of ISE, JSSATEB**



**2022-2023**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING  
JSS ACADEMY OF TECHNICAL EDUCATION**

**JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060**

JSS MAHAVIDYAPEETHA, MYSURU  
**JSS ACADEMY OF TECHNICAL EDUCATION**  
JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that DBMS LABORATORY WITH MINI PROJECT (18CSL58) Report entitled “**AICTE Activity Database Management**” is a Bona-fide work carried out by **ARUN M MIRLE [1JS20IS024]**, **ASHISH B R [1JS20IS025]** in partial fulfilment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2022- 2023.

---

**Signature of the Guide**

**Dr. Sowmya K N**  
**Associate Professor**  
**Dept. of ISE, JSSATEB**

---

**Signature of the HOD**

**Dr. Rekha P M**  
**Professor & HOD**  
**Dept. of ISE, JSSATEB**

# TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>I</b>
<b>CHAPTER1</b>	
<b>INTRODUCTION</b>	
HISTORY OF DBMS	1
OBJECTIVES	2
ORGANIZATION OF THE REPORT	3
<b>CHAPTER 2</b>	
<b>LITERATURE SURVEY</b>	
INTRODUCTION	4
<b>CHAPTER 3</b>	
<b>REQUIREMENT SPECIFICATIONS</b>	
SOFTWARE SPECIFICATION	10
HARDWARE SPECIFICATION	10
USER CHARACTERISTICS	10
FUNCTIONAL REQUIREMENTS	10
NON-FUNCTIONAL REQUIREMENTS	11
USE CASES	11
<b>CHAPTER 4</b>	
<b>SYSTEM DESIGN</b>	
INTRODUCTION TO SYSTEM DESIGN	13
ATTRIBUTES	14
SCHEMA DIAGRAM	15
ER DIAGRAM	16
WIREFRAME	18
<b>CHAPTER 5</b>	
<b>PROJECT IMPLEMENTATION</b>	
INTRODUCTION	23
CREATING TABLES	23
QUERIES	28
HTML CODE	31

## **CHAPTER 6**

### **SYSTEM TESTING**

INTRODUCTION	35
TYPES OF TESTING	35

## **CHAPTER 7**

RESULTS AND DISCUSSIONS	41
-------------------------	----

## **CHAPTER 8**

### **CONCLUSION AND FUTURE ENHANCEMENTS**

CONCLUSION	47
FUTURE ENHANCEMENT	47

## **CHAPTER 9**

### **REFERENCES**

BOOK REFERENCES	48
WEB REFERENCES	48

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So with gratitude, we acknowledge all those whose guidance and encouragement crowned our efforts with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Project Work as a part of our curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekka P. M.** for her co-operation and encouragement at all the moments of our approach

It is our pleasant duty to place on record our deepest sense of gratitude to our respected project guide **Dr. Sowmya K N.** Associate Professor for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all ISE Department Teachers, **Mrs. Sowmya G** mam, our laboratory instructor and non-teaching staff for providing us with their valuable guidance and for being there at all stages of our work.

**ARUN M MIRLE [1JS20IS024]**

**ASHISH B R [1JS20IS025]**

## **ABSTRACT**

Automation is the motto of this fast moving world, and being able to do anything from anywhere is the luxury this generation has provided. Aicte Activity Management system which evolved in Computer management system is used for managing data entries and logging event details. AICTE Activity Management systems incorporate students enrolling for events, viewing organization details and live events and its details.

This project aims at demonstrating the working of these systems with the above mentioned services highly comparable to their real-world implementations. The purpose of this AICTE Activity Management System project is to improve the maintenance and manipulation of the student's data and event log data. The AICTE Activity management system will be used to minimize the time and resource by maintaining the details of the events systemically so that the data can be used in possible quickest time. While the resource which is minimized are workforce, money, papers, etc. The system is user-friendly and will help the administrator.

# CHAPTER 1: INTRODUCTION

## HISTORY OF DBMS:

### I. Early Work

Early Work assumed that files were on tape. Access was sequential and the cost of access grew in direct proportion to the size of the file.

### II. The emergence of Disks and Indexes

As files grew very large, unaided sequential access was not a good solution. Disks allowed for direct access. Indexes made it possible to keep a list of keys and pointers in a small file that could be searched very quickly. With the key and pointer, the user had direct access to the large, primary file.

### III. The Emergence of Tree Structures

As indexes also have a sequential flavor, when they grew too much, they also became difficult to manage. The idea of using tree structures to manage the index emerged in the early 60's. However, trees can grow very unevenly as records are added and deleted, resulting in long searches requiring many disk accesses to find a record.

### IV. Balanced Trees

In 1963, researchers came up with the idea of AVL trees for data in memory. AVL trees, however, did not apply to files because they work well when tree nodes are composed of single records rather than dozens or hundreds of them. In the 1970's came the idea of B-Trees which require an  $O(\log_k N)$  access time where  $N$  is the number of entries in the file and  $k$ , the number of entries indexed in a single block of the B-Tree structure --> B-Trees can guarantee that one can find one file entry among millions of others with only 3 or 4 trips to the disk.

## V. Hash Tables

Retrieving entries in 3 or 4 accesses is good, but it does not reach the goal of accessing data with a single request. From early on, Hashing was a good way to reach this goal with files that do not change size greatly over time. Recently, Extendible Dynamic Hashing guarantees one or at most two disk accesses no matter how big a file becomes.

## OBJECTIVES:

The Objectives of **AICTE ACTIVITY DATABASE MANAGEMENT** are:

1. The main objective of our project is to provide an easy interface for the faculty and students of any college to *integrate* and *organise* all the data related to AICTE activities conducted by college.
2. This project also aims at *providing access* to AICTE ACTIVITY data across various departments in the university.
3. It also aids in the *preservation of EVENTS LOGS* for years to come. In case any faculty wants to *retrieve information* of previously conducted activity, this mini-project will provide an appropriate solution.
4. This project will provide sorting assistance to Admin, sorting can be done through Admin dashboard.



## ORGANIZATION OF THE REPORT

**Chapter 1** the history of DBMS and the objectives. In **chapter 2** we discuss about the literature survey. In **Chapter 3**, we discuss the software and hardware requirements and functional and non-functional requirements to run the above applications. **Chapter 4** gives the idea of the system design. **Chapter 5** gives a clear picture about the project and its actual implementation. In **Chapter 6** we discuss about the system testing. **Chapter 7** discusses about the results and discussions of the program. **Chapter 8** concludes by giving the direction for future enhancement and the **Chapter 9** includes the references.

## CHAPTER 2: LITERATURE SURVEY

### INTRODUCTION

To understand Database Management System (DBMS), it is necessary to know about Database and even before that, about Data. Data is any facts or figures which can be recorded and which has a latent meaning. Database is a collection of such data, more accurately it is a collection of related data. For e.g. a database of countries of the world and their calling code. Here, countries and calling code are related to each other and the database serves a purpose, to provide user with correct calling code. A random collection where, data is neither related nor serve a purpose cannot be called as a database. In summary a database is a collection of logically coherent data, which come together to serve a particular purpose and they emulate some aspect of real world.

Software which is used to manage such databases is called DBMS. For e.g. Oracle, IBM DB2, Microsoft Access, Microsoft SQL Server, Sybase. DBMS software is a collection of programs which facilitate in access, retrieval, security, creation and sharing of database among its users or other applications. Primary Goal of DBMS Software is to provide an efficient and convenient way of storing and retrieving data.

Here are four types of database languages and their uses:

#### 1. Data definition language (DDL)

Data definition language (DDL) creates the framework of the database by specifying the database schema, which is the structure that represents the organization of data. Its common uses include the creation and alteration of tables, files, indexes and columns within the database. This language also allows users to rename or drop the existing database or its components. Here's a list of DDL statements:

- CREATE: Creates a new database or object, such as a table, index or column
- ALTER: Changes the structure of the database or object
- DROP: Deletes the database or existing objects
- RENAME: Renames the database or existing objects

## 2. Data manipulation language (DML)

Data manipulation language (DML) provides operations that handle user requests, offering a way to access and manipulate the data that users store within a database. Its common functions include inserting, updating and retrieving data from the database. Here's a list of DML statements:

- INSERT: Adds new data to the existing database table
- UPDATE: Changes or updates values in the table
- DELETE: Removes records or rows from the table
- SELECT: Retrieves data from the table or multiple tables

## 3. Data control language (DCL)

Data control language (DCL) controls access to the data that users store within a database. Essentially, this language controls the rights and permissions of the database system. It allows users to grant or revoke privileges to the database. Here's a list of DCL statements:

- GRANT: Gives a user access to the database
- REVOKE: Removes a user's access to the database

## 4. Transaction control language (TCL)

- Transaction control language (TCL) manages the transactions within a database.
- Transactions group a set of related tasks into a single, executable task. All the tasks must succeed in order for the transaction to work. Here's a list of TCL statements:
- COMMIT: Carries out a transaction
- ROLLBACK: Restores a transaction if any tasks fail to execute

## MySQL

A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.



Fig 2.1: MySQL logo

## PHP

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.



Fig 2.2: php logo

## CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.



Fig 2.3: css logo

## HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. 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. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items.



Fig 2.4: HTML5 logo

## JAVASCRIPT

JavaScript often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).



Fig 2.5: JAVASCRIPT logo

## Windows 11

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), and is available for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the canceled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and Direct Storage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser, like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via the Microsoft Store and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).



Fig 2.6: WINDOWS 11 logo

## CHAPTER 3: REQUIREMENT SPECIFICATIONS

### SOFTWARE SPECIFICATION

- Project Type: Web-Based Application
- Front-end Tech: HTML, CSS, BOOTSTRAP, JavaScript
- Database Tool: MySQL
- Back-end Tech: PHP
- OS: Windows 8 and above
- Browser: Chrome, Firefox, or Brave
- Software: XAMPP

### HARDWARE SPECIFICATION

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or greater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical.

### USER CHARACTERISTICS

Every user:

- Should be comfortable with basic working of the computer
- Must be literate at the basic level

### FUNCTIONAL REQUIREMENTS

**Functional requirements** are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behaviour under specific conditions.

- The system sends an alert when invalid USN is entered, registration is successful if and only if USN format is valid.



- The system also sends an alert when the password entered by the user does not have at least 1 uppercase, 1 lowercase and 1 digit.
- A filter feature allows the admin to segregate the available data as per the requirement.
- The software automatically validates USN and Phone number against database makes sure it is not repeated.
- Only Admin have the right to view all student and event logs data.

## NON-FUNCTIONAL REQUIREMENTS

**Non-functional requirements**, not related to the system functionality, rather define *how* the system should perform.

- The website pages should load in 3 seconds with the total number of simultaneous users <5 thousand.
- The system should be able to handle 20 million users without performance deterioration.
- The software should be portable. So moving from one OS to other OS does not create any problem.
- Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.

## USE CASES

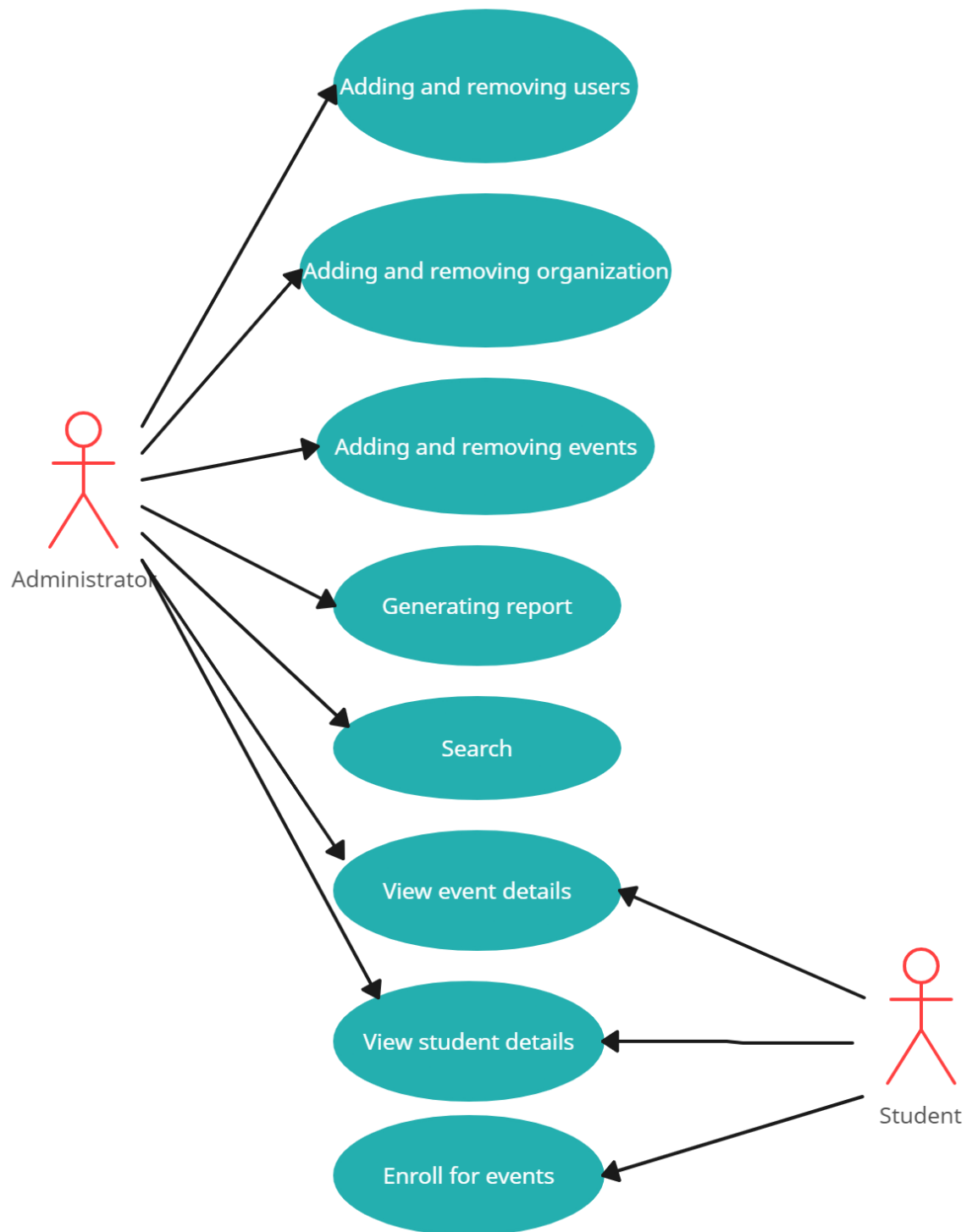
Use cases describe the interaction between the system and external users that leads to achieving particular goals. Each use case includes three main elements:

**Actors.** These are the external users that interact with the system.

**System.** The system is described by functional requirements that define an intended behaviour of the product.

**Goals.** The purposes of the interaction between the users and the system are outlined as goals.

The below figure, Fig 3.1 showcases the use case diagram for Aicte Activity DBMS



**Fig 3.1: Use case diagram of Aicte activity DMBS**

## CHAPTER 4: SYSTEM DESIGN

### INTRODUCTION TO SYSTEM DESIGN

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “how to implement?” It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.

Systems design is the process or art of defining the architecture, components modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

**Database Design** is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

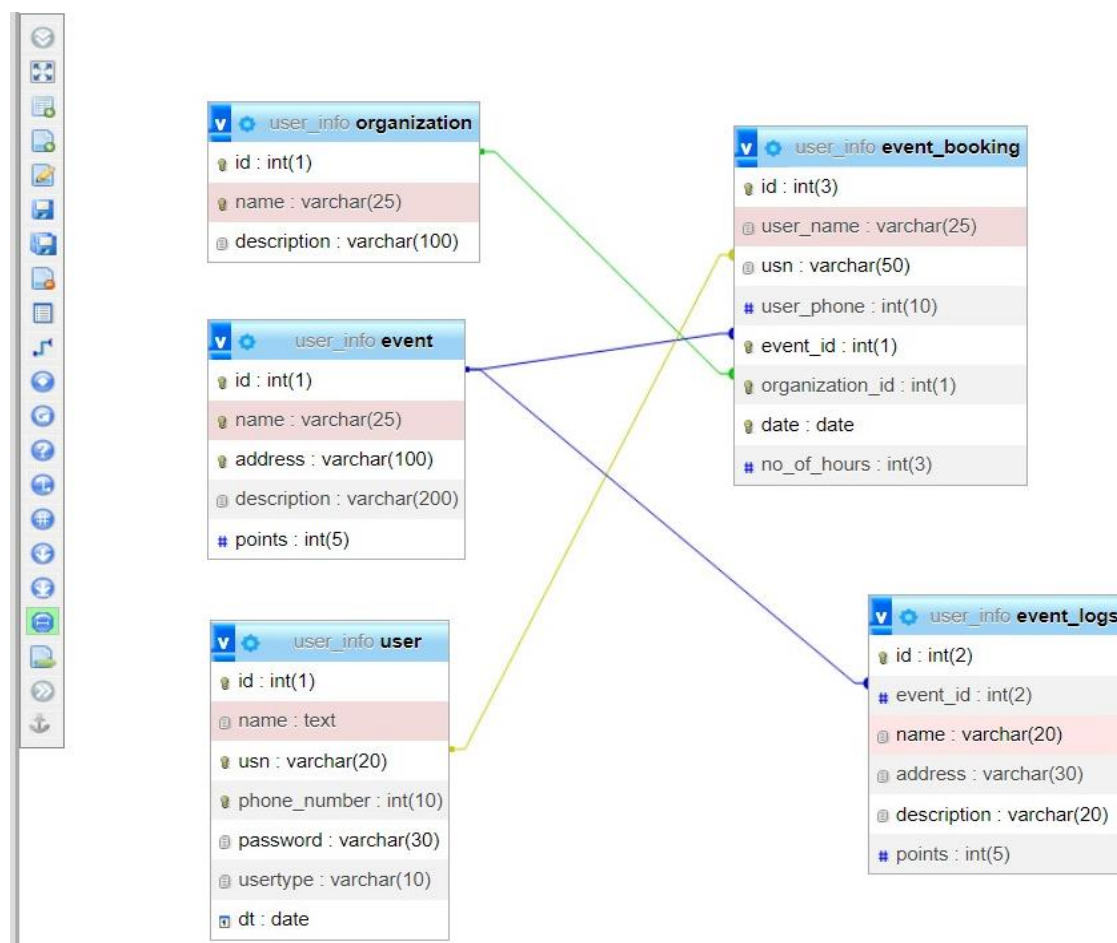
## ATTRIBUTES

Attributes define the properties of a data object and take on one of three different characteristics.

They can be used to:

- Name an instance of data object.
- Describe the instance.

The following Fig 4.1 shows the Relational model for research paper database management.



**Fig.4.1 Displays the relational model**

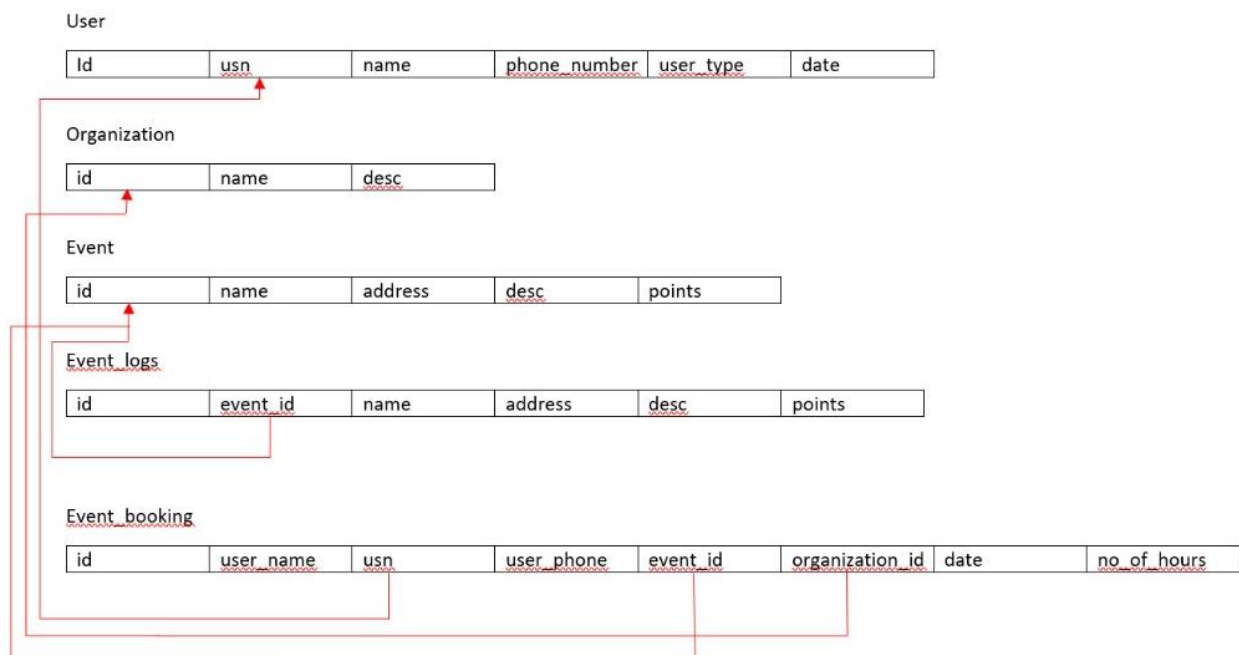
Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc. So we will see what Relational Model is. Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).

## SCHEMA DIAGRAM

A **database schema** is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

A **schema diagram** contains entities and the attributes that will define that **schema**. It only shows us the database design. It does not show the actual data of the database. **Schema** can be a single table or it can have more than one table which is related.

The below figure, Fig 4.2 showcases the schema diagram for Aicte Activity DBMS



**Fig 4.2 Schema Diagram of proposed AICTE activity management system**

A database schema is a structure that represents the logical storage of the data in a database. It represents the organization of data and provides information about the relationships between the tables in a given database. In this topic, we will understand more about database schema and its types. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database. All the various table used are described in the following schema. The necessary Primary key's and the corresponding Foreign keys are also represented.

## ER DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. These symbols are depicted in the figure Fig 4.3; this figure illustrates the symbols in the ER diagram.

**Rectangle:** Represents Entity sets.

**Ellipses:** Attributes

**Diamonds:** Relationship Set

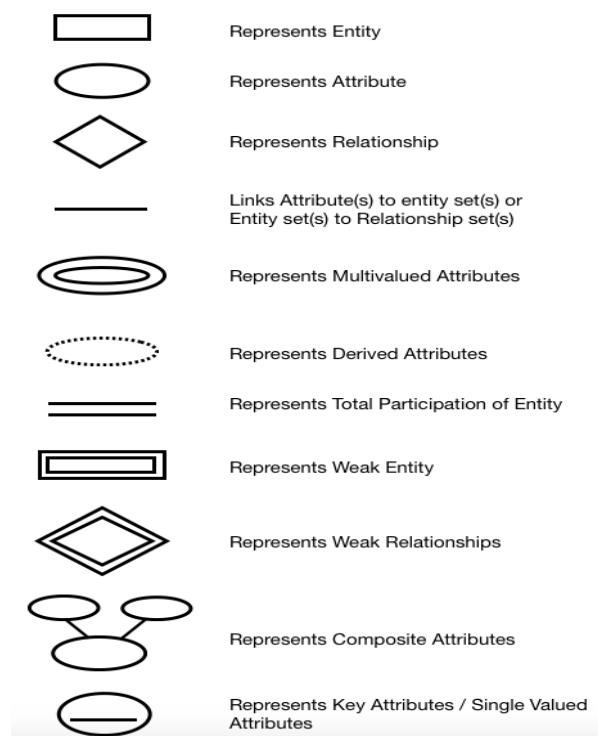
**Lines:** They link attributes to Entity Sets and Entity sets to Relationship Set

**Double Ellipses:** Multivalued Attributes

**Dashed Ellipses:** Derived Attributes

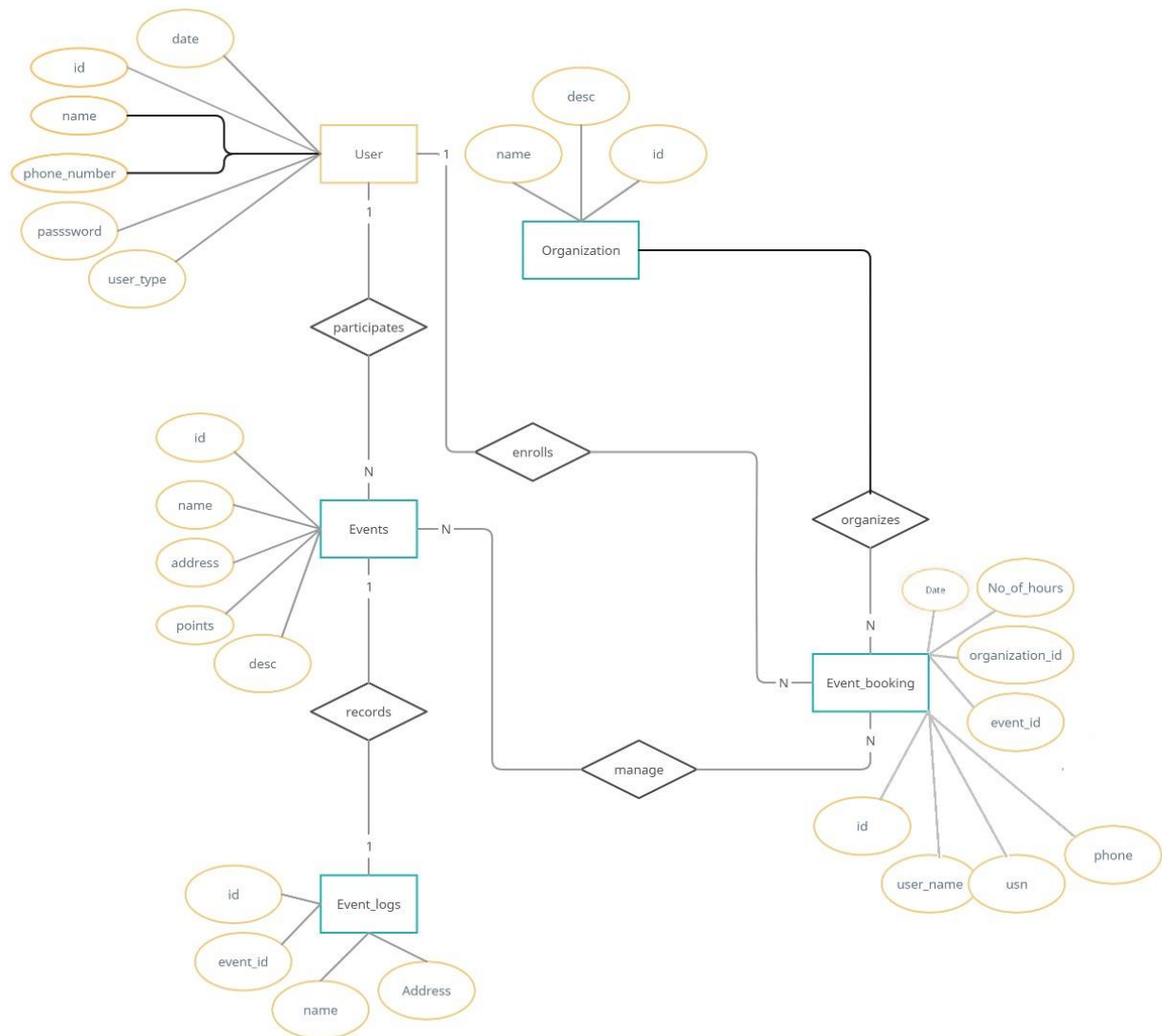
**Double Rectangles:** Weak Entity Sets

**Double Lines:** Total participation of an entity in a relationship set



**Fig 4.3 ER Diagram Symbols**

An entity–relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types. The figure below Fig 4.4 shows the entities and their attributes, the entities relationship with other entities is also illustrated with respective multiplicity.



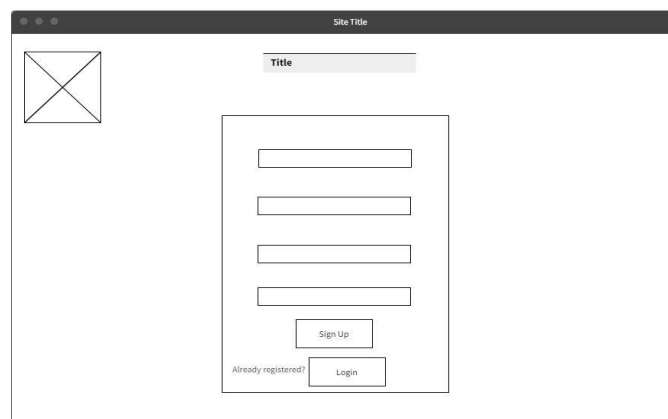
**Fig 4.4 ER Diagram of proposed AICTE activity management system**

## WIREFRAME

A wireframe is a two-dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of content, functionalities available, and intended behaviours. For these reasons, wireframes typically do not include any styling, colour, or graphics. Wireframes also help establish relationships between a website's various templates.

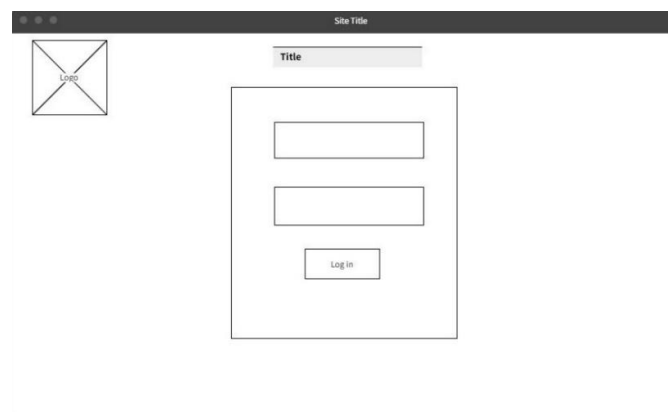
Whenever the project is launched in the local host the homepage will be rendered in the web which will allow the user to login/sign up in the figure , upon login the student can interact with different buttons available in the home page where the particular button will render the enrolment forms which allow the user to interact with the database as shown in the Fig 4.5.

Fig 4.5 is the sign up page where student can Register. Upon Registration user is redirected to login page Fig 4.6.



**Fig.4.5: Signup page**

Fig 4.6 is the login page where student and admin can login using their usn and password, user will be redirected to homepage or admin dashboard.

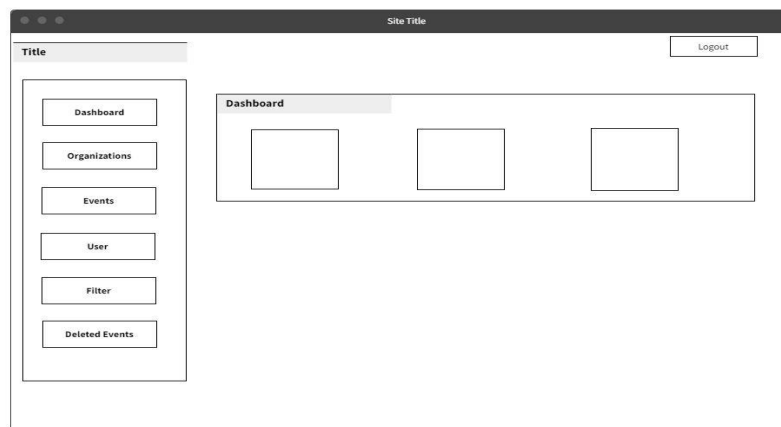


**Fig 4.6: Login page**



## ADMIN POV

Fig 4.7 is admin dashboard page.



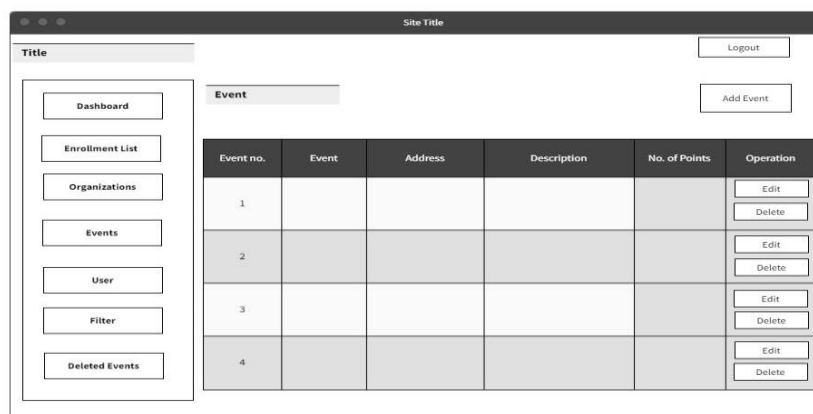
**Fig 4.7: Admin dashboard**

Fig 4.8 is organization page where all the list of organizations are shown



**Fig. 4.8: Admin Organization**

Fig 4.9 represents the events page where it shows all the events



**Fig. 4.9: Admin Events**

Fig 4.10 represents the list of user's page where it shows all the users and their details

#	Name	USN	Phone Number	User Type	Operation
1					Edit Delete
2					Edit Delete
3					Edit Delete
4					Edit Delete

**Fig. 4.10: User page**

Fig 4.11 represents filter where we can filter according to Organizations Events Users.

#	Student Information	Enrollment Information
1		
2		
3		

**Fig 4.11: Filter page**

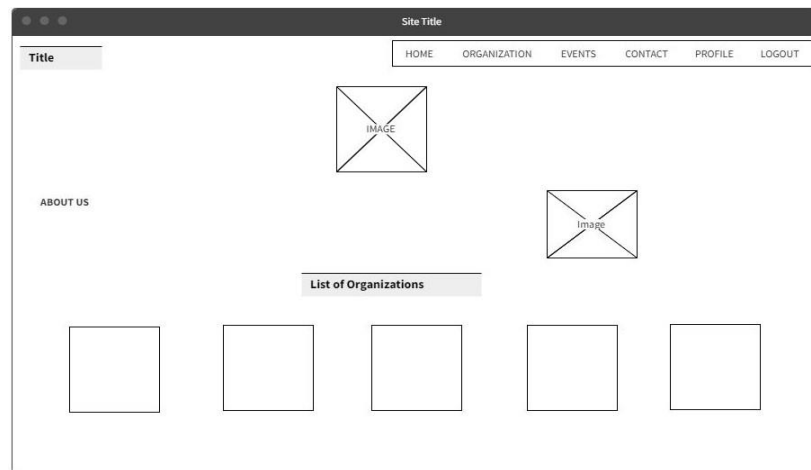
Fig 4.12 represents where it shows the deleted events

Event no.	Event	Address	Description	No. of Points	Operation
1					Restore
2					Restore
3					Restore
4					Restore

**Fig. 4.12: Deleted events page**

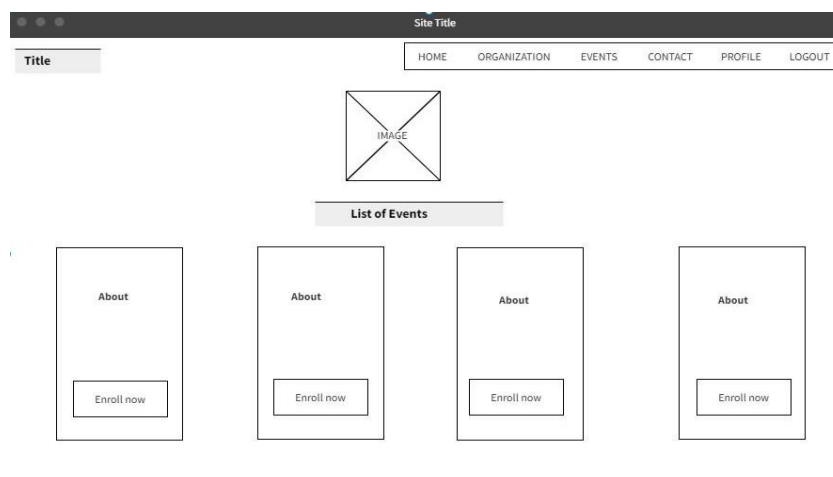
## STUDENT POV

Fig 4.13 represents the home and organizations page of students, upon clicking on events button in menu, student will be redirected to events page Fig 4.14.



**Fig: 4.13 Home page**

Fig 4.14 represents the Events page of students where all the events are shown, upon clicking on enrol student will be redirected to enrol form Fig 4.14.



**Fig: 4.14: Events page**

Fig 4.15 represents enrol form page where student can enrol for a particular event and confirm

**Fig. 4.15: Enrolment page**

Fig 4.16 is a profile page where it displays student's information with delete operation.

#	Organization Type	Event Type	Organization Address	Date	No of Hours	Points	Action
1							Cancel
2							Cancel
3							Cancel
4							Cancel

**Fig 4.16: Profile page**

## CHAPTER 5: PROJECT IMPLEMENTATION

### INTRODUCTION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigating of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The following codes will ensure the complete implementation of our design and the project.

### CREATING TABLES

#### TABLE USER

```
CREATE TABLE `user` (  
  `id` int(1) NOT NULL,  
  `name` text NOT NULL,  
  `usn` varchar(10) NOT NULL,  
  `phone_number` int(10) NOT NULL,  
  `password` varchar(30) NOT NULL,  
  `usertype` varchar(10) NOT NULL,  
  `dt` date NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(1)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 name	text	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 usn	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 phone_number	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	5 password	varchar(30)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6 usertype	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7 dt	date			No	None			Change  Drop  More

```
mysql> select *from user;
```

id	name	usn	phone_number	password	usertype	dt
1	admin	1JS11AD001	123	admin12345	admin	2022-01-01
4	xyz	1JS11XY001	987654322	xyz12345	user	2022-01-12
91	abc	1JS11AB001	1234567890	abc12345	user	2022-04-01
92	AKASH G	1JS20IS010	2147483647	Password10	user	2023-01-31
93	BHARGAV B	1JS20IS035	2147483647	Password35	user	2023-01-31
94	HARISH H	1JS20IS040	2147483647	Password40	user	2023-01-31
95	RAKSHITHA R	1JS20IS055	2147483647	Password55	user	2023-01-31
96	SAVITHA S	1JS20IS060	2147483647	Password60	user	2023-01-31

```
8 rows in set (0.00 sec)
```

Fig:5.1: Selection from the User table

## TABLE ORGANIZATION

```
CREATE TABLE `organization` (
  `id` int(1) NOT NULL,
  `name` varchar(25) NOT NULL,
  `description` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(1)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 name	varchar(25)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 description	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More

```
mysql> select *from organization;
```

id	name	description
1	Rotary Club	Dedicated organization to help locals
2	NGO-N	National Global Organization
3	NGO-R	Retribution
4	WHO	Food shortage manager
5	NCC	Serve Food

Fig:5.2: Selection from the organizations table

## TABLE EVENT

```
CREATE TABLE `event` (
  `id` int(1) NOT NULL,
  `name` varchar(25) NOT NULL,
  `address` varchar(100) NOT NULL,
  `description` varchar(200) NOT NULL,
  `points` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>id</b> 🔑	int(1)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	<b>name</b> 🔑	varchar(25)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>address</b> 🔑	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	<b>description</b>	varchar(200)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	<b>points</b>	int(5)			No	None			Change  Drop  More

```
mysql> select *from event;
+----+-----+-----+-----+-----+
| id | name          | address                | description                | points |
+----+-----+-----+-----+-----+
| 1  | Govt Agara School | Agara main rd, Agara, Bangalore | Teach students from 1st to 7th | 20     |
| 2  | Blood Donation   | #23, Shivaginagar      | Voluntre Blood Donation   | 20     |
| 3  | Swatch Bharath   | Jayanagar, Bangalore.  | Clean Roads                | 20     |
| 4  | Tourism          | Electronic City, Bangalore. | Tourist guide              | 20     |
| 5  | Old age home service | Whitefield, Bangalore. | Help old people            | 20     |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Fig:1.2: Selection from the Event table

## TABLE EVENT\_BOOKING

```
CREATE TABLE `event_booking` (
  `id` int(3) NOT NULL,
  `user_name` varchar(25) NOT NULL,
  `usn` varchar(10) NOT NULL,
  `user_phone` int(10) NOT NULL,
  `event_id` int(1) NOT NULL,
  `organization_id` int(1) NOT NULL,
  `date` date NOT NULL,
  `no_of_hours` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(3)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 user_name	varchar(25)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 usn	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 user_phone	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	5 event_id	int(1)			No	None			Change  Drop  More
<input type="checkbox"/>	6 organization_id	int(1)			No	None			Change  Drop  More
<input type="checkbox"/>	7 date	date			No	None			Change  Drop  More
<input type="checkbox"/>	8 no_of_hours	int(3)			No	None			Change  Drop  More

```
mysql> select *from event_booking;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | user_name | usn | user_phone | event_id | organization_id | date | no_of_hours |
+----+-----+-----+-----+-----+-----+-----+-----+
| 82 | xyz | 1JS11XY001 | 987654322 | 1 | 2 | 2022-03-15 | 20 |
| 86 | xyz | 1JS11XY001 | 987654322 | 2 | 3 | 2022-04-26 | 20 |
| 89 | BHARGAV B | 1JS20IS035 | 2147483647 | 1 | 1 | 2022-12-12 | 20 |
| 90 | AKASH G | 1JS20IS010 | 2147483647 | 2 | 2 | 2023-01-11 | 10 |
| 91 | HARISH H | 1JS20IS040 | 2147483647 | 4 | 5 | 2023-01-31 | 10 |
| 94 | RAKSHITHA R | 1JS20IS055 | 2147483647 | 2 | 2 | 2022-12-12 | 10 |
| 95 | SAVITHA S | 1JS20IS060 | 2147483647 | 3 | 3 | 2022-12-14 | 10 |
+----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Fig:1.4: Selection from the event\_booking table



## TABLE EVENT\_LOGS

```
CREATE TABLE `event_logs` (
  `id` int(2) NOT NULL,
  `event_id` int(2) NOT NULL,
  `name` varchar(20) NOT NULL,
  `address` varchar(30) NOT NULL,
  `description` varchar(20) NOT NULL,
  `points` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	ISSNnumber	int(40)			No	None			Change  Drop  More
<input type="checkbox"/> 2	Titleofpaper	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	Nameoftheauthor	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	Nameofjournal	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	Yearofpublication	int(4)			No	None			Change  Drop  More
<input type="checkbox"/> 6	Links	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 7	fid	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More

```
mysql> select *from event_logs;
+----+-----+-----+-----+-----+-----+
| id | event_id | name          | address   | description   | points |
+----+-----+-----+-----+-----+-----+
| 12 |      1 | HighSchool Teaching | Kengeri   | Teach students |    20 |
| 15 |      2 | Blood donation    | Uttarahalli | Blood Donation |    20 |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Fig:1.1: Selection from the Event\_logs table

## QUERIES

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update, triggers and procedures.

### USER TABLE

**Query:**

```
SELECT * FROM `user` WHERE `usn` = '$usn'
```

**Description:** In this query, we aim to fetch the data using WHERE, the query fetches data from user table based on *usn*, where *usn* is the primary key in user table.

**Query:**

```
SELECT * FROM `user` WHERE `phone_number` = '$phone_number'
```

**Description:** In this query, we aim to fetch the data using WHERE, the query fetches data from user table based on *phone\_number*, where *phone\_number* is another primary key in user table.

### ORGANIZATION TABLE

**Query:**

```
SELECT * FROM `organization` WHERE `name` = '$name'
```

**Description:** In this query, we aim to fetch the data using WHERE, the query fetches data from organization table based on its *name*, where *name* is the primary key in organization table.

### EVENT TABLE

**Query:**

```
SELECT * FROM `event` WHERE `name` = '$name'
```

**Description:** In this query, we aim to fetch the data using WHERE, the query fetches data from event table based on its *name*, where *name* is the primary key in event table.

## EVENT BOOKING

**Query:**

```
SELECT * FROM `event_booking` where `organization_id`='$organization_id'
```

**Description:** In this query, we aim to fetch the data using WHERE, the query fetches data from event table based on *organization\_id*, where *organization\_id* is a foreign key in event\_booking table

## TRIGGERS

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

**Syntax:**

```
CREATE TRIGGER [TRIGGER_NAME]
[BEFORE | AFTER]
{INSERT | UPDATE | DELETE}
ON [TABLE_NAME]
[FOR EACH ROW]
[TRIGGER_BODY]
```

### Trigger for EVENT

In this trigger we aim to create a record of the deleted events, upon deletion of an event, that particular event will be inserted in to event\_logs table, which can be restored later using the Deleted events page through Admin access.

```
CREATE TRIGGER `tr_delete` AFTER DELETE ON `event`
FOR EACH ROW INSERT into event_logs(event_id,name,address,description,points)
VALUES(old.id, old.name, old.address, old.description, old.points)
```

### Trigger for EVENT\_LOGS

In this trigger we aim to restore a record from the event\_logs table, upon clicking on restore button in Deleted events page, that particular event will be inserted in to events table again.

```
CREATE TRIGGER `del_logs` AFTER DELETE ON `event_logs`  
FOR EACH ROW INSERT into event  
VALUES(old.event_id,old.name, old.address,old.description,old.points)
```

### STORED PROCEDURES

#### Syntax:

```
DELIMITER &&  
CREATE PROCEDURE procedure_name  
[[IN | OUT | INOUT] parameter_name datatype [, parameter datatype]] ]  
BEGIN  
    Declaration_section  
    Executable_section  
END &&  
DELIMITER ;
```

#### Procedure for USER

This procedure is used to fetch the details of the student searched by the admin. This procedure is called when the admin selects a particular student's usn. This procedure compares the usn of the student in event\_booking table, and then returns the corresponding details of that student.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getBookings`  
(IN `usnid` VARCHAR(25))  
SELECT * FROM event_booking WHERE usn = usnid and event_booking.id = id$$
```

## HTML CODE

*index.html*

```
<!DOCTYPE html>
<html lang="en">
<?php
    $insert = false;
    $server = "localhost";
    $username = "root";
    $password = "";

    $connection = mysqli_connect($server, $username, $password);
    mysqli_select_db($connection, "user_info");
    if(!$connection){
        die("connection to this database failed due to" . mysqli_connect_error());
    }
    if(isset($_POST["submit"])){
        $name = isset($_POST['name']) ? $_POST['name'] : '';
        $usn = isset($_POST['usn']) ? $_POST['usn'] : '';
        $phone_number = isset($_POST['phone_number']) ? $_POST['phone_number'] : '';
        $passwd = isset($_POST['password']) ? $_POST['password'] : '';

        $sql_usn="SELECT * FROM `user` WHERE `usn` = '$usn'";
        $sql_phone="SELECT * FROM `user` WHERE `phone_number` = '$phone_number'";
        $db_usn = mysqli_query($connection, $sql_usn);
        $db_phone = mysqli_query($connection, $sql_phone);

        if(mysqli_num_rows($db_usn)!=0){
            $usn_error = "usn already taken";
        }
        elseif(mysqli_num_rows($db_phone)!=0){
            $phone_error = "Phone number already taken";
        }
        else{
            $sql = "INSERT INTO `user`(`name`,`usn`,`phone_number`,`password`,`usertype`,`dt`)
VALUES ( '$name', '$usn', '$phone_number','$passwd', 'user' , current_timestamp());";
            $result = mysqli_query($connection, $sql);
            // $insert = true;
            if($result)
            {
                echo "<script>alert('Registration successful!');</script>";
                echo "<script> window.location = 'login.php'</script>";
            }
            //header("location: login.php");
        }
        $connection->close();
    }
?>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome</title>

    <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css?v=<?php echo time(); ?>">
    <div class="header">
        <h1>AICTE ACTIVITY</h1>
        <p style="margin-top: -20px;line-height:1;font-size:30px;">Database Management System
Project</p>
    </div>
```

```

</head>
<body>
<div class="logo">
  
</div>
<div class="page">
  <div class="container m-auto">
    <h3>REGISTRATION</h3>
    <form action="" method="post">
      <div class="row">
        <div class="col">
          <hr class="mb-3">
          <label for="name"><b>Name</b></label>
          <input class="form-control" type="text" name="name" id="name" placeholder="Enter
your name" required>
          <label for="usn"><b>USN</b></label>
          <div <?php if (isset($usn_error)): ?> class="form_error" <?php endif ?> >
            <input class="form-control" type="usn" name="usn" id="usn" placeholder="Enter
your usn" required>
            <?php if (isset($usn_error)): ?>
              <span><?php echo $usn_error; ?></span>
            <?php endif ?>
          </div>
          <label for="phone_number"><b>Phone Number</b></label>
          <div <?php if (isset($phone_error)): ?> class="form_error" <?php endif ?> >
            <input class="form-control" type="phone" name="phone_number" id="phone_number"
placeholder="Enter phone number" required>
            <?php if (isset($phone_error)): ?>
              <span><?php echo $phone_error; ?></span>
            <?php endif ?>
          </div>
          <label for="password"><b>Password</b></label>
          <div class="pass">
            <input class="form-control" type="password" name="password" id="password"
placeholder="Enter a password" required>
            <i><span class="material-icons" id="visibilityBtn">
              visibility
            </span></i> </div>
            <hr class="mb-3">
            <input type="submit" id="signup" name="submit" value="Sign up!"
onclick="validate(usn)">
          </div>
        </div>
        <div>Already registered? Login <a href="login.php">here</a></div>
      </form>
    </div>
  </div>
  <br>
  <br>
  <br>
  <br>
  <div>
    </div>
    <?php
      unset($usn_error);
      unset($phone_error);
      unset($error);
    ?>
  </div>
<script type="text/javascript">
  function validate(usn)
  {
    var error="";

    var pattern=/^[1-4][A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{3}$/;
  }

```

```

        if (!usn.value.match(pattern))
        {
            error = "USN is invalid!";
            window.location.href="index.php";
        }

        if (error.length == 0)
            alert("USN is valid!");
        else
            alert(error);
    }
</script>

<script>
    $(document).ready(function() {

        jQuery.validator.defaults({
            errorPlacement: function(error, element) {
                error.appendTo('#invalid_' + element.attr('id'));
            }
        });

        var validator = $("#login_form").validate({
            rules: {
                user_email: {
                    required: true,
                    email: true
                },
                password: {
                    required: true
                }
            }
        });
    });
</script>

<script>
$.validator.addMethod("phone IN", function (phone_number,element) {

phone_number    =    phone_number.replace(/\s+/g, "");    return    this.optional(element)    ||
phone_number

.length > 9 &&

phone_number.match(/^([6-9]{1}\d{9}){1}?$/));

}, "Please specify a valid phone number.");
</script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script type="text/javascript">

const visibilityBtn = document.getElementById("visibilityBtn");

visibilityBtn.addEventListener("click", toggleVisibility);

function toggleVisibility(){
    const password = document.getElementById("password");
    if(password.type === "password"){
        password.type = "text";
    }else {
        password.type = "password";
    }
}

```

```
}
</script>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>

</body>

</html>
```

## CODE TO ESTABLISH CONNECTION BETWEEN FRONTEND AND DATABASE

*connection..php*

```
<?php

$host="localhost";
$user="root";
$password="";

if(!isset($_SESSION))
{
    session_start();
}
$data=mysqli_connect($host,$user,$password);
mysqli_select_db($data, "user_info");
if($data===false)
{
    die("connection error");
}

?>
```



## **CHAPTER 6: SYSTEM TESTING**

### **INTRODUCTION**

Testing plays a vital role in the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Once program code has been developed, testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals, that is conducted tests to uncover errors and ensure that defined input will produce actual results that agree with required results. Broadly speaking, there are at least three levels of testing: unit testing, integration testing, and system testing

### **TYPES OF TESTING**

#### **Unit testing**

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other. Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.

Sl.no	Test Case	Expected Result	Test Result
1	Enter valid usn and password and click login (both admin and user)	Software should display the dashboard.	Successful
2	Enter invalid usn or password and click login (both admin and user)	Software should display invalid usn or password.	successful

**Table 6.1: Unit testing case for user validation**

## Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

## System Testing

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff. The tables below from Table 6.2-6.12 describes all the testing.

## Registration form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the interface between Registration and Database	Enter the registration details on click on Sign up	The user need to fill the particular details, the details should be validated if the user already exists a message window will appear as alert, otherwise the data should be updated to the database and should render the signup form.	Successful

Table 6.2: Testing case for Registration form validation

## Login form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the interface between Login and Dashboard or Homepage	Enter the login credentials (usrn and password) and click on login	To be Directed to Admin or User Dashboard page as per credentials entered.	Successful

Table 6.3: Testing case for Login form validation

## Homepage form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the interface link between the Homepage and logout module	When present on the Homepage page, If the admin or user wants to logout click on logout	To be directed to the Login page of the Aicte Activity website.	Successful

Table 6.4: Testing case for Logout

## Enrol form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the interface between Enrol page and profile	Enter the details of the enrolment and click on Confirm	The profile needs to be updated with the enrolled event. If the user enrolls for another event on the same date, an alert should appear, to avoid the enrolment of the user for that event.	Successful

Table 6.5: Testing case for Enrol form validation

## Dashboard form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the interface link between the Dashboard and logout module	When present on the Dashboard page, If the admin or user wants to logout click on logout	To be Directed to Admin or User Dashboard page as per credentials entered.	Successful

Table 6.6: Testing case for Dashboard Page

## Enrolment list form

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the Delete operation in Enrolment list page	In Enrolment List page click on Delete Button to delete the particular enrolment details	That row of details should be deleted and should no longer appear in Enrolment list page. The corresponding event will also be deleted for that student	Successful
2	To check the operation of Print button in Enrolment list page	In Enrolment List page click on print button, user should be redirected to Activity report page with another print button, upon clicking that, a window will appear with print options.	The user should be redirected to Activity report page and the print window should appear with save and print options.	Successful

Table 6.7: Testing case for Enrolment List Page

## Organization page from

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the Delete operation	In Organization page click on Delete Button to delete that particular organization	That row of details should be deleted and should no longer appear in Organization page.	Successful
2	To check the Edit operation	In Organization page click on edit button, the user will be redirected to edit organization page where the user can change Name and Description fields	The user should be redirected to edit organization page and the changes made should appear in the List of organizations	Successful
3	To check the Add operation	In organization page click on Add organization button, Enter the Name and Description of new organization.	When the Add organization button is clicked, a window should appear with Name and Description fields to be entered	Successful

Table 6.8: Testing case for Organization Page

## Events page from

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the Delete operation	In Events page click on Delete Button to delete that particular event	That row of details should be deleted and should no longer appear in Events page.	Successful
2	To check the Edit operation	In Events page click on edit button, the user will be redirected to edit event page where the user can change Event name, Address, Short Description and Points fields	The user should be redirected to edit organization page and the changes made should appear in the List of events.	Successful
3	To check the Add operation	In Events page click on Add event button, Enter the Event name, Address, Short Description and Points fields of the new event	When the Add event button is clicked, a window should appear with Event name, Address, Short Description and Points fields to be entered	Successful

Table 6.9: Testing case for Organization Page

## User page from

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the Delete operation	In User page click on Delete Button to delete that particular user	That row of details should be deleted and should no longer appear in User page.	Successful
2	To check the Edit operation	In Users page click on edit button, the user will be redirected to edit user page where the user can change Name, USN, Phone number and Password fields.	The user should be redirected to edit user page and the changes made should appear in the List of users	Successful
3	To check the Add operation	In organization page click on Add organization button, Enter the Name , USN, Phone number and Password of new user.	When the Add user button is clicked, a window should appear with Name, USN, Phone number and Password fields to be entered	Successful

Table 6.10: Testing case for Organization Page

Filter page from

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the Organization Type, Event Type and User Filter option	The user should select a particular organization type, event type or user, and click on go to carry out filter operation	The records with that filter selections should appear in the table below.	Successful

**Table 6.11: Testing case for Filter Page**

Deleted events page

SL. NO	Test Case Objective	Test case Description	Expected Results	Output
1	To check the restore operation	In User page click on Restore Button to restore that particular event.	The restored records should reappear in the events page.	Successful

**Table 6.12: Testing case for Deleted Events Page**



## CHAPTER 7: RESULTS AND DISCUSSIONS

**SIGNUP:** When the user loads the webpage, this is the first page, the user if new, enters the new details for registration, particular form will be shown as shown below in Fig 7.1 where the new user can enter the name, usn, phone number, password which will be stored in user table.

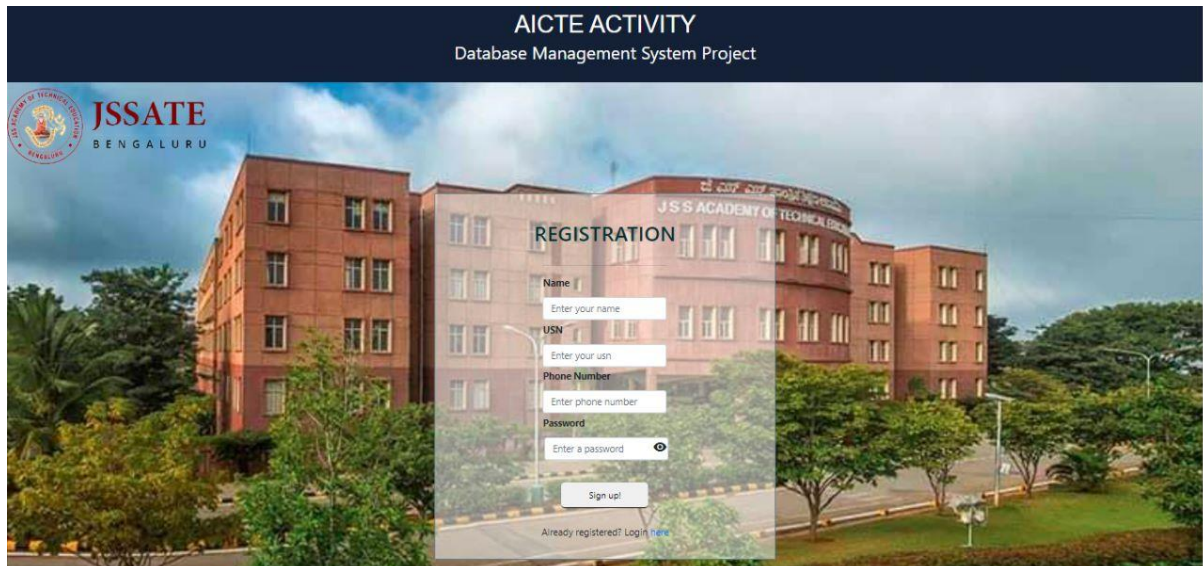


Fig 7.1: Signup page [Landing page]

**LOGIN:** If the user data already exist in the database upon clicking the signup button on the login the user can login to the system using the password and usn the signup page looks like the Fig 7.2.

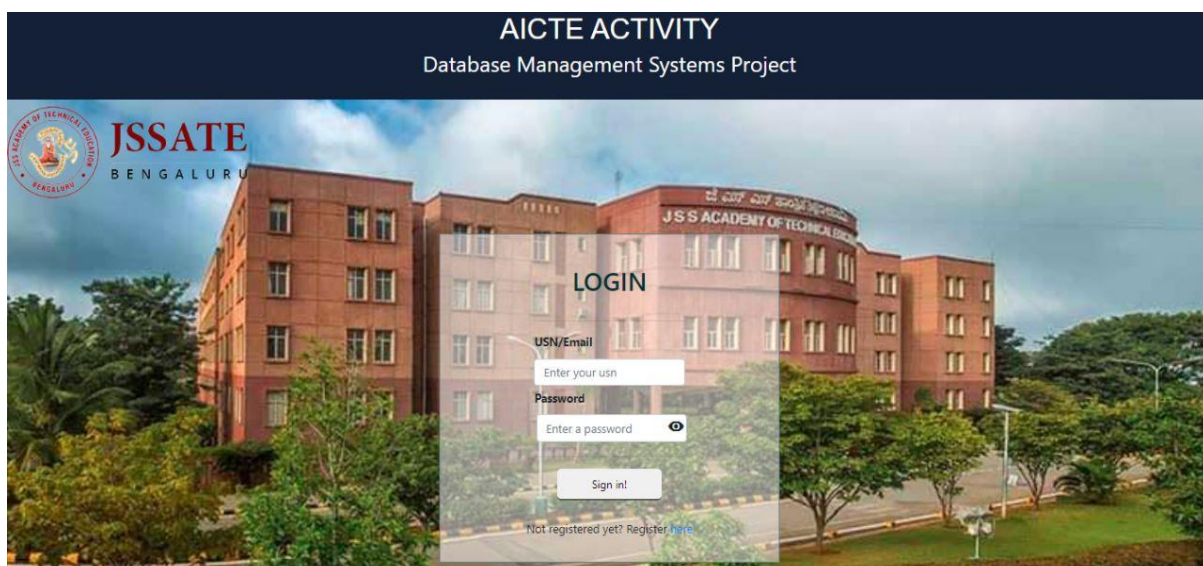
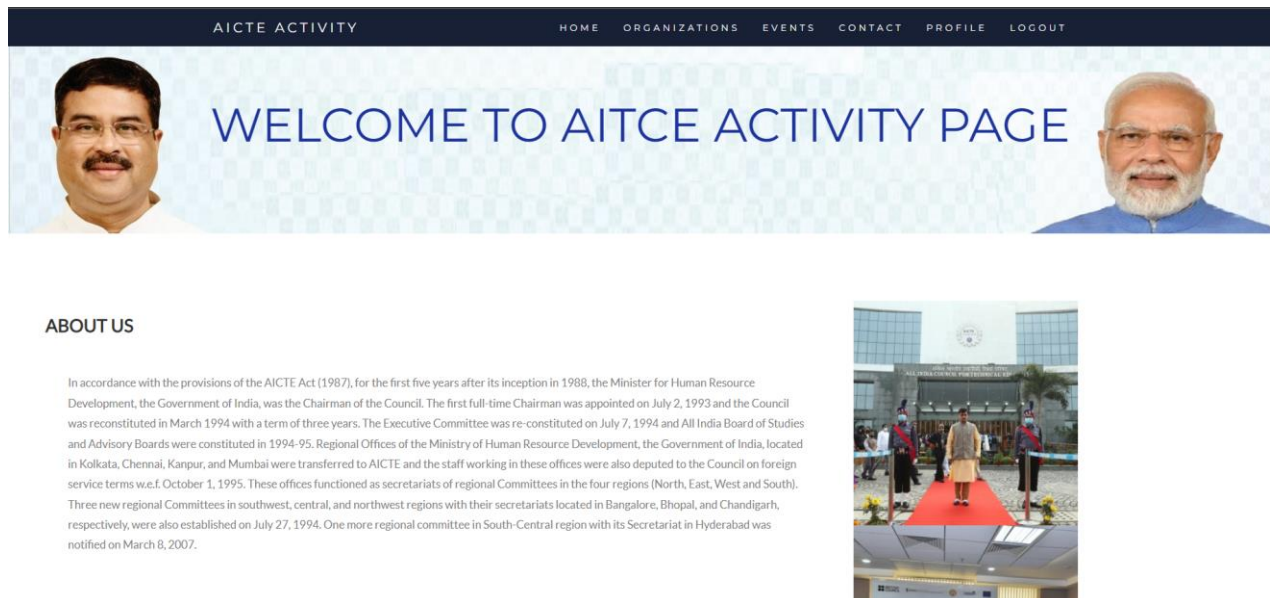
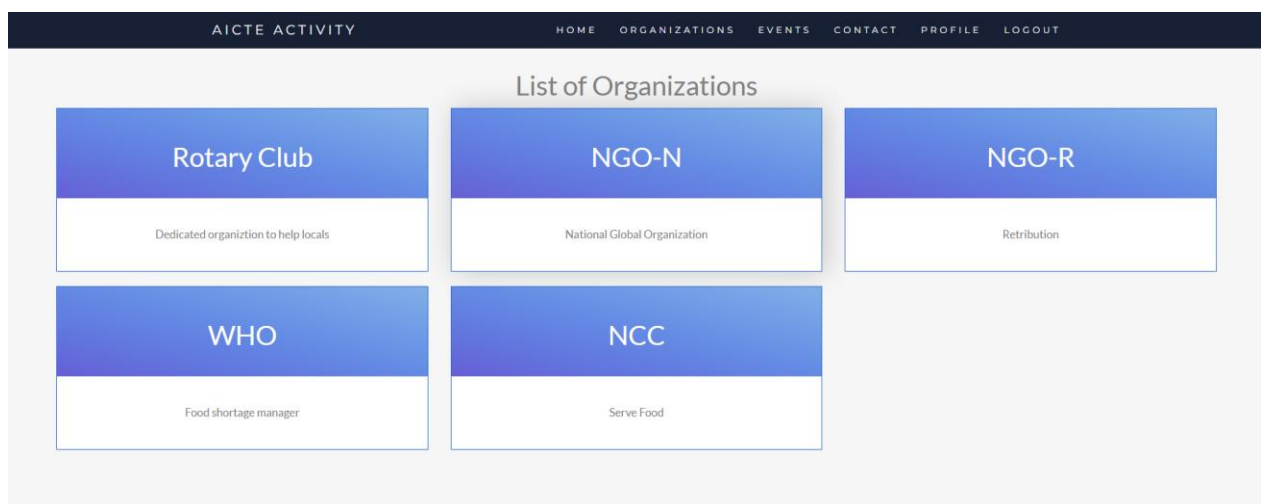


Fig 7.2: Login page

HOME PAGE: The home page will be rendered which will be as shown in Fig 7.3 which consists of different button which will be used to perform the different task and redirect to different pages. This page also contains the List of organizations as shown in Fig 7.4.



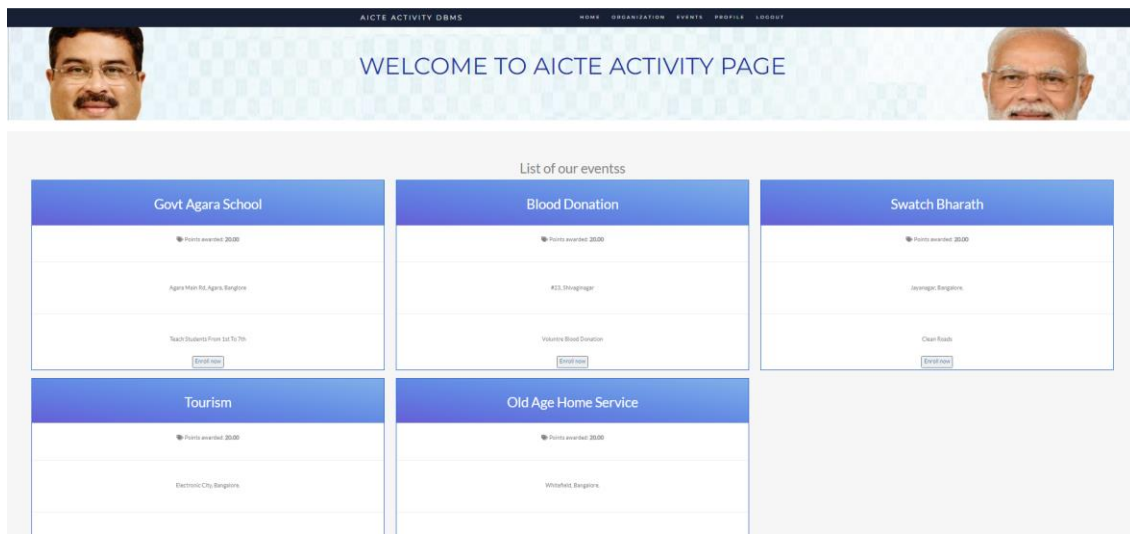
**Fig 7.3: Landing page of Aicte Activity data management**



**Fig 7.4: List of organizations page**



EVENT PAGE: Fig 7.5 shows the events page, with events modules to which a student can enrol.



**Fig 7.5: List of events page**

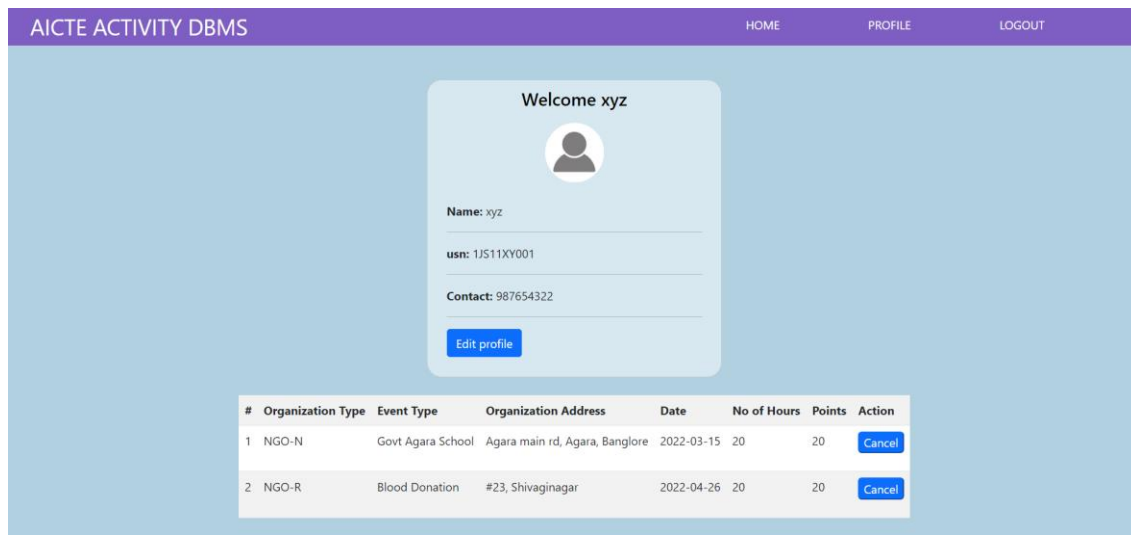
ENROL FORM PAGE: Fig 7.6 shows the enrol form which the student needs to fill.

The screenshot shows the 'Enroll for events by filling your details here' form. The form has a purple header with 'AICTE ACTIVITY' and a 'Home' link. The form fields are:

- Name: Enter Name
- USN: Enter usn
- Phone: Enter phone
- Organization Type: Rotary Club
- Date: dd-mm-yyyy
- No of Hours: Enter no of hours
- Confirm button

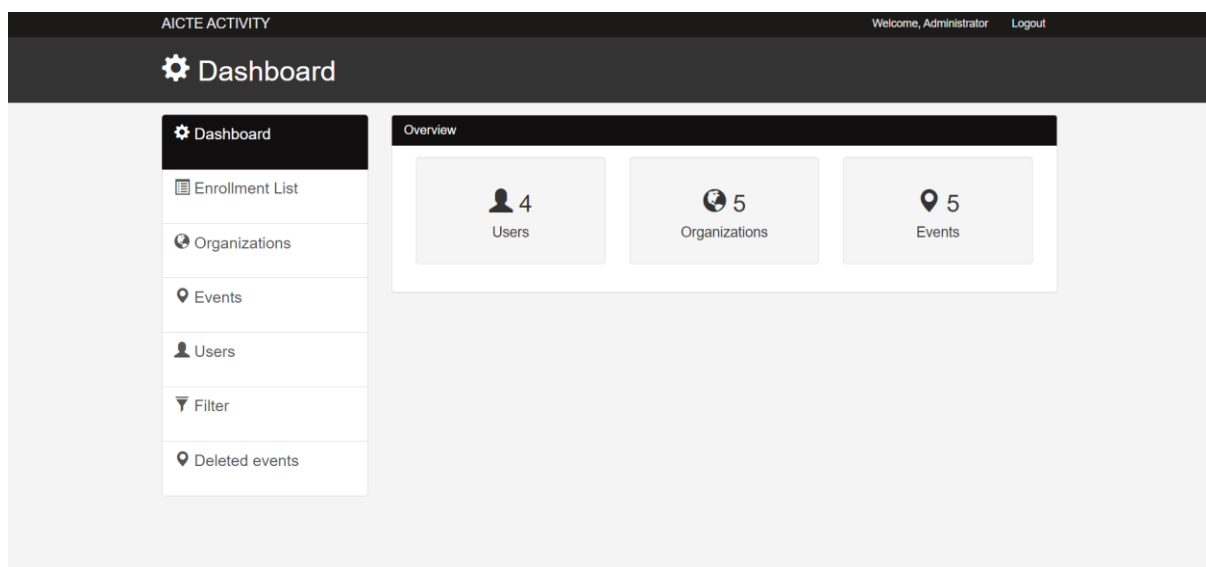
**Fig 7.6: Enrol page**

**PROFILE PAGE:** Whenever the user clicks the Profile button available in the home page the Profile page will be rendered which will be as shown in Fig7.7, which will display all the details of the students in the database where user can delete the particular event data from the database.



**Fig 7.7: Profile page**

**ADMIN DASHBOARD PAGE:** Fig 7.8 depicts the Admin dashboard with all its menu options and functionality. Here the can access the Enrolment list and list of organizations, events and users and also the feature of filtering.



**Fig 7.8: Admin dashboard**

ENROLMENT PAGE: Fig 7.9 shows the list of enrolments of the students and its details. It has a print button which will print Activity report.

#	Student Information	Enrollment Information	Operation
1	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	ORGANIZATION NAME: NGO-N EVENT NAME: Govt Agara School EVENT ADDRESS: Agara main rd, Agara, Bangalore DATE: 2022-03-15 NUMBER OF HOURS: 20 points: 20	Edit Delete
2	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	ORGANIZATION NAME: NGO-R EVENT NAME: Blood Donation EVENT ADDRESS: #23, Shivaginagar DATE: 2022-04-26 NUMBER OF HOURS: 20 points: 20	Edit Delete

Print

**Fig 7.9: Enrolment list page (Admin)**

ACTIVITY REPORT PAGE: When the user clicks on Print, we are redirected to activity report print page (Fig 7.10), which is the set print format. And upon clicking print, the print window will appear.

## ACTIVITY REPORT

#	Student Information	Enrollment Information
1	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	ORGANIZATION NAME: NGO-N EVENT NAME: Govt Agara School EVENT ADDRESS: Agara main rd, Agara, Bangalore DATE: 2022-03-15 NUMBER OF HOURS: 20 Points: 20
2	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	ORGANIZATION NAME: NGO-R EVENT NAME: Blood Donation EVENT ADDRESS: #23, Shivaginagar DATE: 2022-04-26 NUMBER OF HOURS: 20 Points: 20
3	NAME: HARISH H usr: 1JS20IS040 CONTACT: 2147483647	ORGANIZATION NAME: NCC EVENT NAME: Tourism EVENT ADDRESS: Electronic City, Bangalore. DATE: 2023-01-31 NUMBER OF HOURS: 10 Points: 20

Print

Fig 7.10: Enrolment list page (Admin)

**FILTER PAGE:** Feature of filtering as per organization, event, usr type. It gives access for the admin to filter as per his needs.

**Filters**

Organization Type:

Event Type:

User:

#	Student Information	Enrollment Information
1	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	organization ID: 2 event ID: 1 DATE: 2022-03-15 No of Hours 20
2	NAME: xyz usr: 1JS11XY001 CONTACT: 987654322	organization ID: 3 event ID: 2 DATE: 2022-04-26 No of Hours 20

Fig 7.11: Filter page with organization type, event type and user (Admin)

## **CHAPTER 8: CONCLUSION AND FUTURE ENHANCEMENTS**

### **CONCLUSION**

AICTE ACTIVITY MANAGEMENT SYSTEM successfully implemented based on online data filling which helps us in administrating the data user for managing the tasks performed by students. The project successfully used various functionalities of Xampp and php and also create the fully functional database management system for online portals. Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.

With the theoretical inclination of our syllabus it becomes very essential to take maximum advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “AICTE Activity Management System” was one of these opportunities.

It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

### **FUTURE ENHANCEMENT**

1. The scope for this project is to aiding in Aicte activities across various colleges to share their findings among each other.
2. The platform can be hosted on online servers to make it accessible worldwide.
3. Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.
4. Distribute the loads of the system.
5. Features like notifying all the users of upcoming Conferences and various such technical seminars can be integrated with this project to carter a larger audience.

## CHAPTER 9: REFERENCES

### BOOK REFERENCES

1. Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. An Introduction to Database Systems” by Bipin Desai.
3. Database System Concepts” by Abraham Silberschatz and S Sudarshan.
4. “Database Management Systems” by Raghu Ramakrishnan.

### WEB REFERENCES

1. W3schools
2. <https://stackoverflow.com/questions>
3. [www.tutorialspoint.com](http://www.tutorialspoint.com)
4. [www.javatpoint.com](http://www.javatpoint.com)
5. <https://www.php.net/docs.php>
6. <https://dev.mysql.com/doc/>