
1. INTRODUCTION

There is an outgrowing need for electronic accessories in a world where there is great necessity speed efficiency and perfection of work. Information factor of efficiency access to electronic devices and relevant information is usually considered in the setting up of a relational database for a local network. Intranet that is a private network set up by an organization or company that reassemble the www (World wide Web) but which is not accessible by external users.

Basically, systems are implemented for facilitating complex manual processes and that is exactly what we are trying to achieve. System is implemented as per user requirement such as a manufacturing concern may install a plant for easing out manual processes. We have sought help from computer programming for automation of manual registration system. With the introduction of computers, every aspect of our lives has been revolutionized. When used judiciously, computers can help us save time, secure our personal information, access the required information whenever and wherever required.

We have developed an Online Student fest Registration System for easily managing the semester registration process for the student in an institution. Ours is an advisory based system. The events are assigned according to the skill set and industry requirements. Hence, in current scenario, automated system is required for fest registration of students.

Features and Functionalities

1. Registration:

This module is available only for user to register for events giving his/her name and other details required and also select the events.

2. Registered students:

Displays the student details and total fee amount is calculated.

3. Header icons:

This links to the respective social media page of the institution.

2. LITERATURE SURVEY

It gets to a stage when the data are too much to be managed in the minds of the elders. And so in order to store all the new information, humanity invented the technology of writing and then great scholars like Aristotle warned that the invention of the alphabet would lead to the subtle but total demise of the creativity and sensibility of humanity, data began to be stored in voluminous data repositories called books. As we know, eventually books capsulated with great speed and soon whole communities of books migrated to the first real “database” libraries. Unlike previous versions of data warehouses (i.e People and books) that might be considered the australopitheaic of the database lineage, libraries crossed over into the modern day species, though they were incredibly primitive of course over into libraries introduced.

The first attempt at information storage and retrieval followed traditional lives and metaphors. The first system were based on discrete files in a virtual library. In this file oriented system, a bunch of files would be stored on a computer and could be accessed by a computer operator. Files of achieved data were called table “ because they looked like record and column in the data. The archival data were called tables because they look like tables used in traditional file keeping, rows in the table were called “Record and columns were called “Field”.

These type of database are usually used to track real-time information for example, a financial company might have an operational database used to track cash transactions that is as customers make payment from an online store an operational database can be used to keep tracks of how much cash is left and the customers draw his credit line.

Traditionally, database ran on large powerful mainframes for business applications. You will probably have heard of such packages as Oracle 10g or Microsoft SQL server, for example. However, with the advent of small powerful personal computer database have become readily usable by the average computer user. MySQL open source Microsoft Access and Borland's Dbase proprietary are a few popular PC-based engines around. More importantly for our focus, database have quickly become integrated to the design, development and service offered by websites.

This data access methodology makes the relational model, a lot different from and better than the earlier database model because it is a much simpler model to understand. This is probably the remaining reason for the popularity of relational database systems today. Another benefit of the relational system is that, it provides extremely useful tools for database administration. Essentially, tables cannot only store actual data but they can also be used as the table means for generating meta-data (data about the table and field names which form the database structure, access rights to the database, integrity and data validation rules etc).

3. ANALYSIS AND REQUIREMENT SPECIFICATION

3.1. Purpose

The programming language in use will be HTML, PHP, JAVA SCRIPT, while the blackened database is Access and VB.net. The choice is as a result of its flexibility when integrated with PHP, HTML helps in the web and interface design. The program was designed in models which is the structure it bears. The source and object code is seen at the appendix column.

In this project, the data sites is assumed small and will probably need a number of submenu pages that users enter from a general category listing on the home page. In complex sites with multiple topic areas, it is not practical to burden the home pages with dozens of links. The page grows too long to load in a timely manner, and its sheer complexity may be off putting to many users, providing a submenu page for each category will create mini-home pages for each section the site.

For specialized details submenus it is recommended that frequent users link there directly, in this way the submenus will become alternative home page in “sub-sites” oriented to a specific audience.

1. Registration
2. Registered students
3. Social media links

Documentation is undoubtedly necessary in every new system development cycle. It is the detailed information about a systems design specification, its internal workings and its functionality. It generally involves written and other visual information about an application system.

3.2. Scope

Data notwithstanding of its size cannot be too large to process. However, a guided approach is adopted to stay within limits for easy comprehension.

To end this, the study will focus on the student together with its accompanying web interface, students fest registration details and academic results. In its most common form, the wide application of computers in accessing remote files make way for adoption of more superior of making information available to the millions of users. And it also exposes the researcher to the core about the difficulties of database programming for the web, an all-round technology but because of the following constraints, the study may not be completed.

3.3. Functional Requirements

3.3.1. General Requirement

- Time constraints and finance
- The college or the organizers does not have enough necessary software and extended network of computers.
- It lacked the projected operational hardware
- Unavailability of power supply to test and implement the developed work

3.4 Non Functional Requirements

3.4.1 Hardware Requirements:

| | |
|-----------|------------------------|
| Processor | Intel Pentium or above |
| RAM | 2GB above |
| Hard disk | 10GB or above |

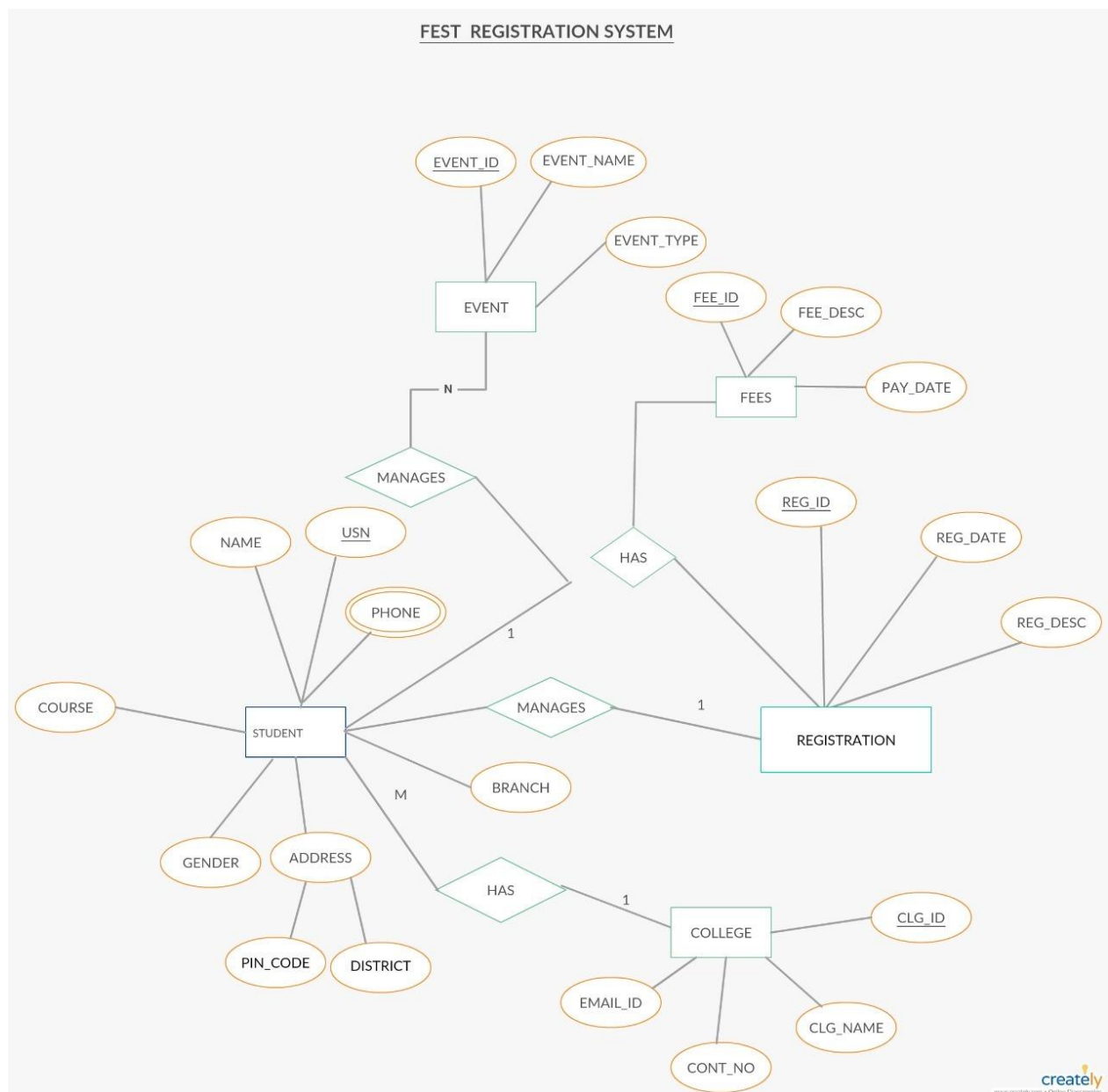
3.4.2 Software Requirements:

| | |
|------------------|---|
| Operating System | Windows |
| Front end | HTML ,CSS, JAVA SCRIPT |
| Language | ASP.NET(C#) |
| Database | MS-SQL |
| Web Browser | Google Chrome, Mozilla, Internet Explorer |

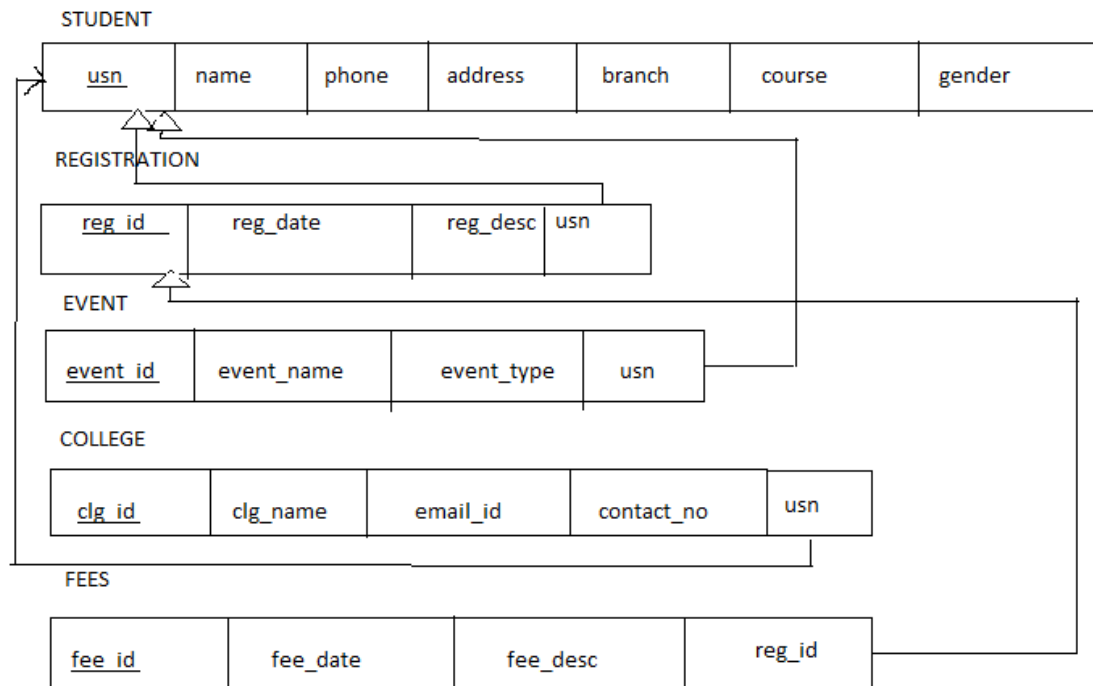
4. DESIGN

While designing the web application portal a three tier architecture for application development was followed. The presentation tier occupies the front end design of the application. It relates to every entity with which the user interacts. It accepts user inputs and actions, and then sends this information to the data tier through the application tier for further processing. The student fest registration portal accepts input in the form of student's personal detail, student's registration details, event details etc.

4.1. ER Diagram



4.2. Schema Diagram



5. IMPLEMENTATION

Top down approach was followed for the implementation of the project. The top-level management formulates the strategy and passes it on to the subordinates with instructions to execute the task.

In the implementation stage all the theoretical details were converted to working functionalities making sure that the user requirements are met.

5.1. Implementation of Table Creation & Insertion

Creating the tables in SQL:

To create the above tables, we first need to create a database to work in. We can do this using the command,

```
CREATE DATABASE fest;
```

Now we can create the tables using the command,

```
Create table student
```

```
(  
  usn varchar(10) primary key,  
  Name text,  
  address varchar(20),  
  phone bigint,  
  pin int,  
  district text,  
  gender char(1),  
  branch varchar(10),  
  emailed varchar(20),  
  course varchar(10),  
  dob date  
);
```

```
Create table college
```

```
(  
  Usn int primary key,  
  College_name text,  
  Foreign key(usn) references student(usn)  
);
```

```
Create table event1
```

```
(  
  Usn int primary key,  
  event text,  
  Foreign key(usn) references student(usn)  
);
```

```
Create table registration1
```

```
(  
  Usn int,  
  Pid int primary key,
```

```
Reg_date date,  
Foreign key(usn) references student(usn)  
);  
Create table fees  
(  
Fee int,  
Pid int primary key,  
fee_date date,  
Foreign key(pid) references registration1(pid)  
);
```


5.2. Implementation of Trigger / Stored Procedure

In this Fest registration system stored procedure and view has been implemented. They are:

- Student details(view)
- Total sum of fee(Stored Procedure)

Implementation of view:

Create view register1

Column names usn,name,college_name,event,pid,reg_date,fee

Select s.usn,s.name,c.college_name,e.event,r.pid,r.reg_date,f.fee

From student s,college c,event1 e,fees f,registration1 r

Where s.usn=c.usn and s.usn=e.usn and s.usn=r.usn and r.pid=f.pid;

Implementation of Procedure:

Create procedure `totalsum`;

Create definer=`root`@`localhost` Procedure`totalsum`() notdeterministic

Contains sql sql security definer

select SUM(fee) from fees;

6. RESULT

6.1. Sample Output (Snapshots)

Fest Registration Form

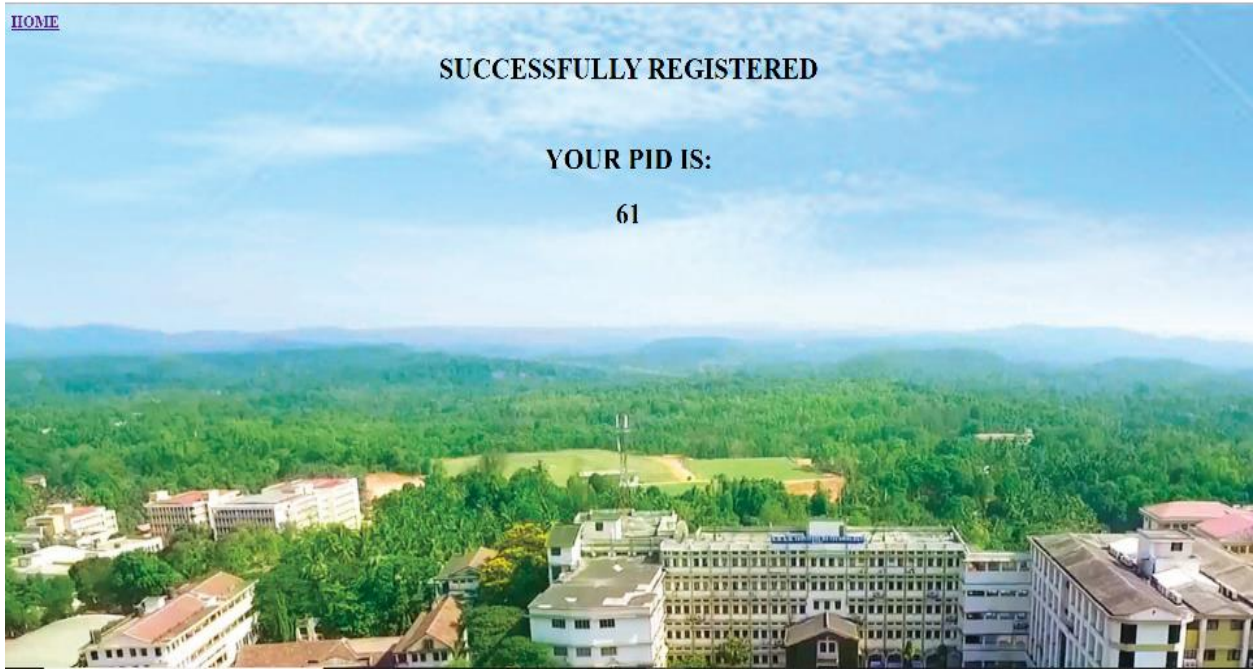
| | |
|--------------------------------------|--|
| Name | <input type="text"/> |
| Usn | <input type="text"/> |
| Address | <input type="text"/> |
| Sex | <input type="radio"/> Male <input type="radio"/> Female |
| BRANCH | <input type="text" value="select.."/> |
| Course | <input type="text" value="select.."/> |
| College Name | <input type="text"/> |
| District | <input type="text" value="select.."/> |
| PinCode | <input type="text"/> |
| EmailId | <input type="text"/> |
| DOB | <input type="text" value="mm/dd/yyyy"/> |
| Events | <input type="checkbox"/> face painting <input type="checkbox"/> battle of bands <input type="checkbox"/> solo dance <input type="checkbox"/> code studio <input type="checkbox"/> classical dance <input type="checkbox"/> fashion show |
| MobileNo | <input type="text"/> |
| Registration date | <input type="text" value="mm/dd/yyyy"/> |
| Fee | <input type="text"/> |
| Fee_Date | <input type="text" value="mm/dd/yyyy"/> |
| <input type="button" value="Reset"/> | <input type="button" value="Submit Form"/> |

[HOME](#)

SUCCESSFULLY REGISTERED

YOUR PID IS:

61



REGISTERED STUDENTS

[HOME](#)

| USN | NAME | COLLEGE | EVENT NAME | PID | REGISTERED DATE | FEE |
|------------|-----------|---------|---------------|-----|-----------------|-----|
| 4nm17is062 | moksha | NMAMIT | face painting | 51 | 2018-09-08 | 100 |
| 4nm17is077 | rakshitha | NMAMIT | face painting | 59 | 2018-09-08 | 100 |
| 4nm17is048 | kavana | NMAMIT | coding | 60 | 2018-09-08 | 100 |
| 4nm17is073 | prathibha | NMAMIT | solo dance | 61 | 2018-09-08 | 100 |

TOTAL AMOUNT IS:

400

7.CONCLUSION

Implementation of the proposed system will reduce the workload of all those involved as the data can be now managed with proper authentication and authorizations instead of being hard copied and accessible to everyone. This system will largely save the precious time details, organizers and faculty incharge ,coordinator. Instead of explicitly signing every document; they just have to acknowledge entries online with the click of a mouse. All the technologies i.e. PHP, Apache and MySQL used for current system design are open source and hence freely available for download. PHP provides a strong platform for creating the visual front-end of the web application and PHP combined with HTML provides a very flexible development environment. For the purpose of fulfilling Web Server requirements XAMPP was used, which is again Open Source and is supported across multiple platforms. In order to maintain visual consistency jQuery was used for simpler implementation of certain features. For constant testing, analysis and execution needs, Firefox and Google Chrome web-browsers were used. With a combination of all these technologies we were able to create a web application environment that is efficient and consistent enough.

Online application of the whole system helps easy access to the system anywhere. Physical presence of the student is not required. The time taken for process completion is now largely reduced. After registration the database is automatically updated at the end of process completion removing the hassle for department officials who had to enter the data manually. As the database is managed through MySQL, data duplication is eliminated and thereby reducing chances of error. Also data can be now be easily retrieved, edited and printed whenever required. Authentication based access proves to be more secure than manual system. Also, database access is authorised and cannot be viewed or edited by unauthorised personnel. So, this automated and computerised system is safe, fast and user friendly.

8. REFERENCES

1. http://www.w3schools.com/howto_css_login_form.asp
2. http://www.roseindia.com/css/css3_button.asp
3. www.computerscijournal.org
4. <https://www.academia.edu>
5. Database Management Systems, Ramakrishna and Gehrke, 3rd Edition, 2014, McGraw Hill