**CHAPTER 1**

**INTRODUCTION**

* 1. **COLLEGE MANAGEMENT SYSTEM**

This College Management System project developed using specific programming languages. The main aim of this project is to develop an online website which covers all the details of college i.e.; Student attendance details, Teacher details, Course details, Marks details, etc. Admin is the Super user of this project. All the record stores in MySQL Database. The proposed software will also reduce the cumbersome paperwork, manual labor as well as communication cost**.**

**Present State:**

* No Software present at all.
* Time consumed in accessing the records of the students.
* Manual creation of lists of eligible and not eligible students.
* More time taken to registration of students on registration forms.

**After implementation of project:**

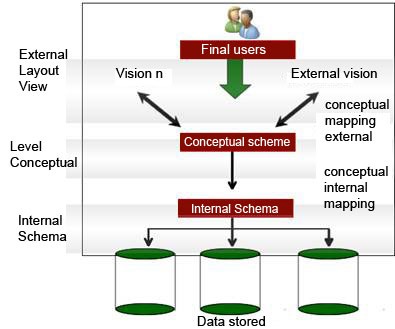
* Easy to use GUI.
* Easy to maintain data of Students and Lecturers.
* Report of students according to need.
* Easy to maintain database and secured storage.
  1. **DBMS**

A database management system (DBMS) is system software for creating and managing [databases](http://searchsqlserver.techtarget.com/definition/database). The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage [data](http://searchdatamanagement.techtarget.com/definition/data).

A DBMS makes it possible for end users to create, read, update and delete [data](http://searchdatamanagement.techtarget.com/definition/data) in a database. The DBMS essentially serves as an interface between the [database](http://searchsqlserver.techtarget.com/definition/database) and end users or [application programs](http://searchsoftwarequality.techtarget.com/definition/application-program), ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database [engine](http://whatis.techtarget.com/definition/engine) that allows data to be accessed, locked and modified and the database [schema](http://searchsqlserver.techtarget.com/definition/schema), which defines the database’s logical structure. These three foundational elements help provide [concurrency](http://searchoracle.techtarget.com/definition/concurrent-processing), security, [data integrity](http://searchdatacenter.techtarget.com/definition/integrity) and uniform administration procedures. Typical database administration tasks supported by the DBMS include [change management](http://searchcio.techtarget.com/definition/change-management), performance monitoring/tuning and [backup](http://searchstorage.techtarget.com/definition/backup) and [recovery](http://searchstorage.techtarget.com/definition/recovery). Many database management systems are also responsible for automated [rollbacks](http://searchsqlserver.techtarget.com/definition/rollback), restarts and recovery as well as the [logging](http://whatis.techtarget.com/definition/log-log-file) and [auditing](http://searchcio.techtarget.com/definition/audit-trail) of activity.

The DBMS is perhaps most useful for providing a centralized view of data that can be accessed by multiple users, from multiple locations, in a controlled manner. A DBMS can limit what data the end user sees, as well as how that end user can view the data, providing many views of a single database schema. End users and software programs are free from having to understand where the data is physically located or on what type of storage media it resides because the DBMS handles all requests.



**Fig 1.1: DBMS Architecture**

**1.3 XAMPP MySQL**

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (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 and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an 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 as well. XAMPP is regularly updated to the latest releases of [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB), [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl). The term XAMPP is an apparent acronym. This is regularly updated to the regular releases of Apache, MariaDB, Perl and PHP.



**Fig 1.2: XAMPP**

* 1. **PROJECT AIM**

Main aim in developing **College Management System** is to provide an easy way not only to automate all functionalities of a college, but also to provide full functional reports to top management of college with the finest of details about any aspect of college.

**CHAPTER 2**

**LITERATURE SURVEY**

The existing online airline booking systems come with many banners of advertisements and pop-ups on the sides of the website that it hinders with the entire booking process. It is known that a booking is not finalized till the transaction of payment is authorized and confirmed. Such have been instances where the user books their ticket provisionally but fails to go through with the transaction. Such tickets are not booked however the reversal of their bookings is not reflected instantly instead take an average of 20 minutes.

Existing system allows only those bookings that are confirmed 5 hours before the departure of the flight. These happenings invite many customer care calls towards the company. To attend these queries, man hours are put on the line.

Many systems crash during the peak hour of their local time. The servers are not enough to handle the load then. Too many users make the page navigations slow, hindering the overall number of bookings that go through till the end.

**CHAPTER 3**

**PROPOSED SYSTEM**

The proposed system towards online College management System is better. Unlike many existing systems, there are multiple servers installed to run the website. This reduces almost all possibilities for a website to hang as a result of server crash.

• When the admissions for the college are open, the student can take admissions. Here the admin can add the details of the students who is taking the admission. The admin can also add the details of the teachers if there is an entry of any new teacher to the college. The admin can also add the faculty details and the details of the marks scored by the student.

• If there are any changes that has to be made in the details of any module, then the admin can update it immediately.

Overall, the proposed system is robust as it caters to all the needs of present-day users expect from an online entry.

.

**CHAPTER 4**

**SYSTEM REQUIREMENT**

**4.1 Hardware Requirements**

|  |  |  |
| --- | --- | --- |
| * PROCESSOR | : | INTEL P-4 BASED SYSTEM |
| * PROCESSOR SPEED | : | 2.0 GHz |
| * RAM | : | 256MB-512MB |
| * HARD DISK | : | 40GB-80GB |

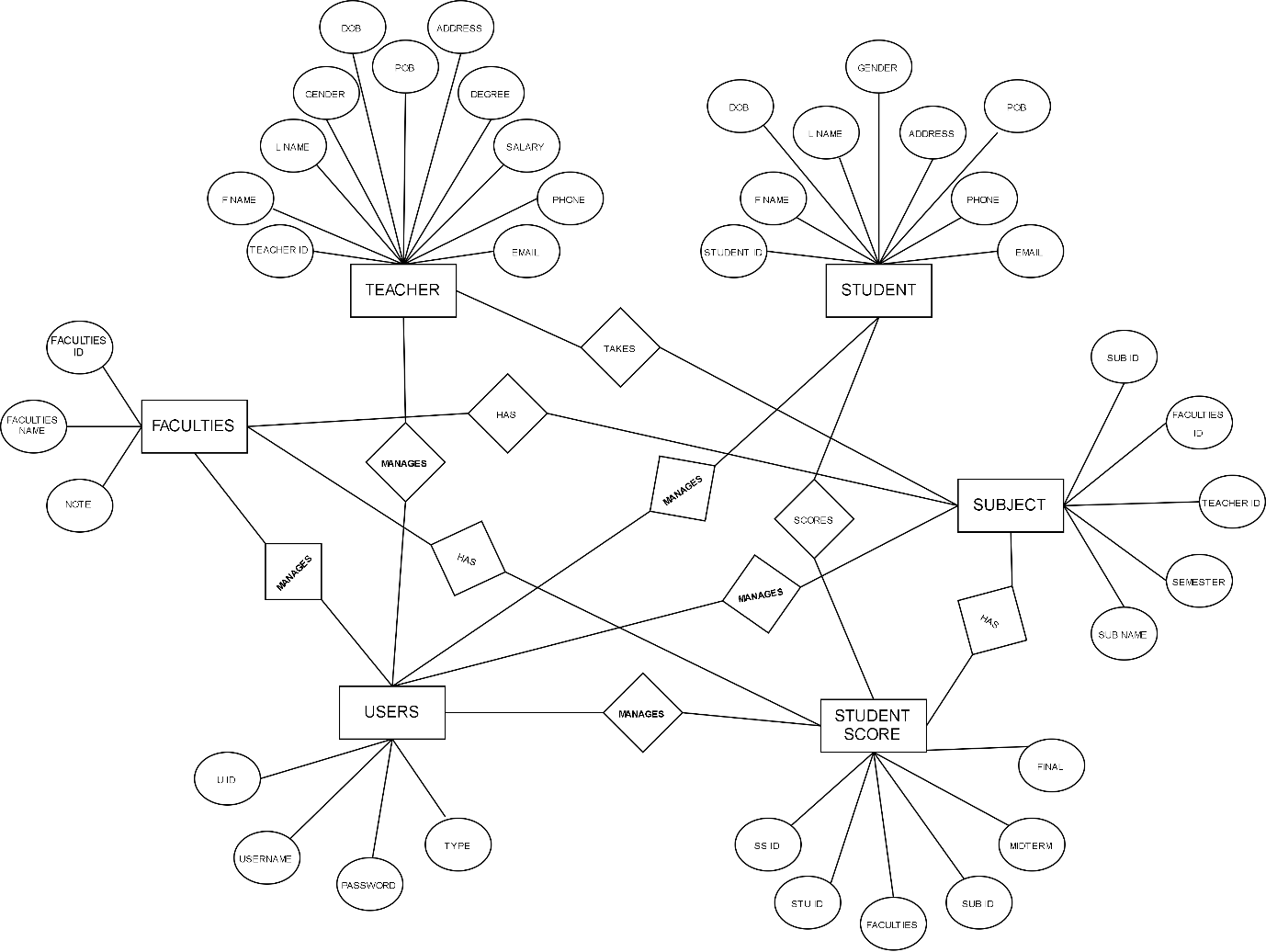
**4.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| * OPERATING SYSTEM | : | WINDOS/LINUX |
| * LANGUAGE | : | PHP, HTML/CSS, JQUERY, JAVASCRIPT |
| * BROWSER | : | GOOGLE CHROME/FIREFOX |

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 ER DIAGRAM**



**Fig 5.1: ER diagram**

**5.2 SCHEMA DIAGRAM**

**Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| U\_id | Username | Password | Type | Note |

**Faculties**

|  |  |  |
| --- | --- | --- |
| Facultiies\_Id | Faculties\_Name | Note |

**Teachers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Teacher\_Id | Name | Gender | Degree | Salary |

**Students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stu\_Id | Name | Gender | Phone | Email |

**Subject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sub\_Id | Faculties\_Id | Teacher\_Id | Semester | Sub\_Name |

**Student Score**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stu\_Id | Faculties\_Id | Sub\_Id | Midterm | Final |

**Fig 5.2: Schema Diagram**

**CHAPTER 6**

**IMPLEMENTATION**

The proposed system of Courier Management System consists of:

* Student module
* Teacher module
* Faculties module
* Subject module
* Score module

**Student module**

The student module consists of the student information. In this module, if the admin wants to add the details of a student who is taking admission then he can enter the details like student name, phone number, address, email-id, date-of-birth, gender and a note if there is any.

**Algorithm**

Step1: start

Step2: login

Step3: enter the student name, phone, email address, gender and address

Step4: enter any note if needed

Step5: click on Register

Step6: stop

**Teacher module**

The office module consists of the teacher information. This module includes teacher name, gender, Date-of-birth, address, qualification, salary, marital status, phone number, email address and a note if there is any.

**Algorithm**

Step1: start

Step2: enter the details of teacher which include teacher name, gender, Date-of-birth, address, qualification, salary, marital status, phone number, email address

Step3: click on Register

Step4: stop

**Faculty module**

The faculty module consists of Faculty information. This module includes faculty name, and a note if there is any.

**Algorithm**

Step1: start

Step2: enter the faculty name and a note if there is any

Step3: click on Register

Step4: stop

**Subject module**

The subject module consists of the details of the subject which includes faculty’s name, teacher’s name, semester, and the subject name and a note if there is any.

**Algorithm**

Step1: start

Step2: enter the faculty’s name, teacher’s name, semester, and the subject name and a note

Step3: click on Add Now

Step4: stop

**Score module**

The score module consists of Scores information. This module includes student name, faculty name, subject name, score in mid-term, score in final exam and a note if there is any.

**Algorithm**

Step1: start

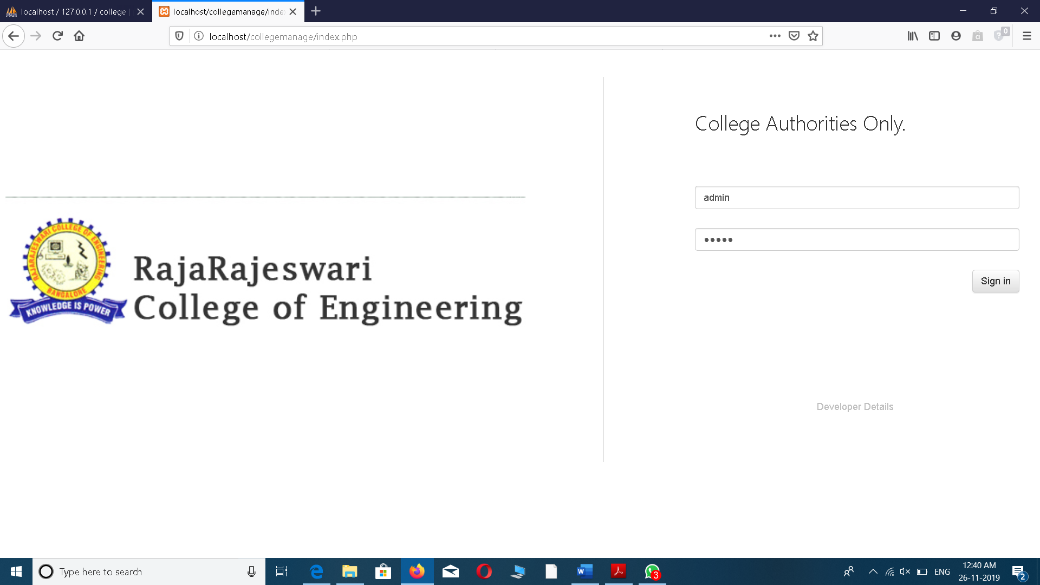
Step2: enter the student name, faculty name, subject name, score in mid-term, score in final exam

Step3: click on Add Now

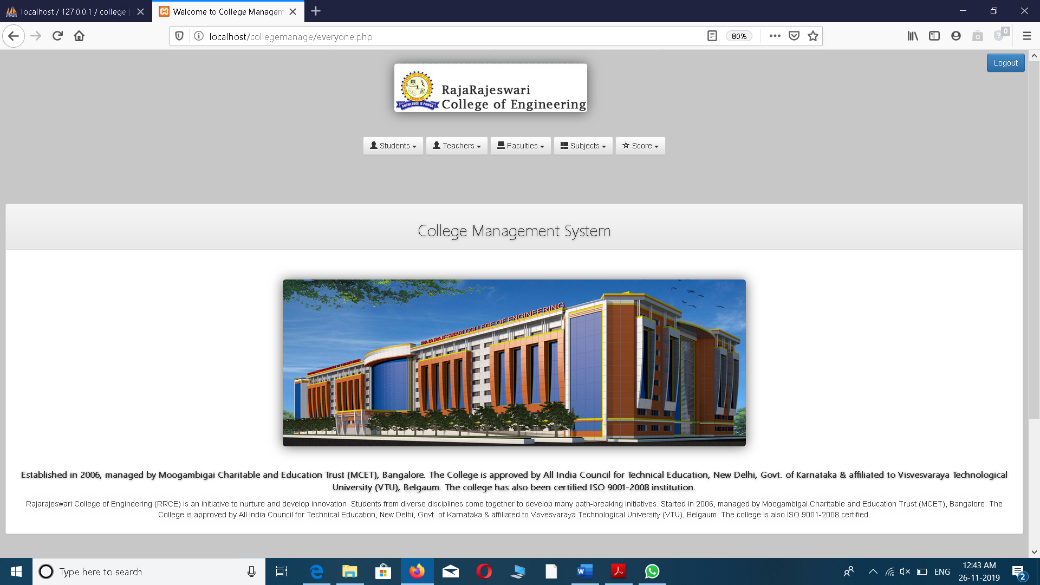
Step4: stop

**CHAPTER 7**

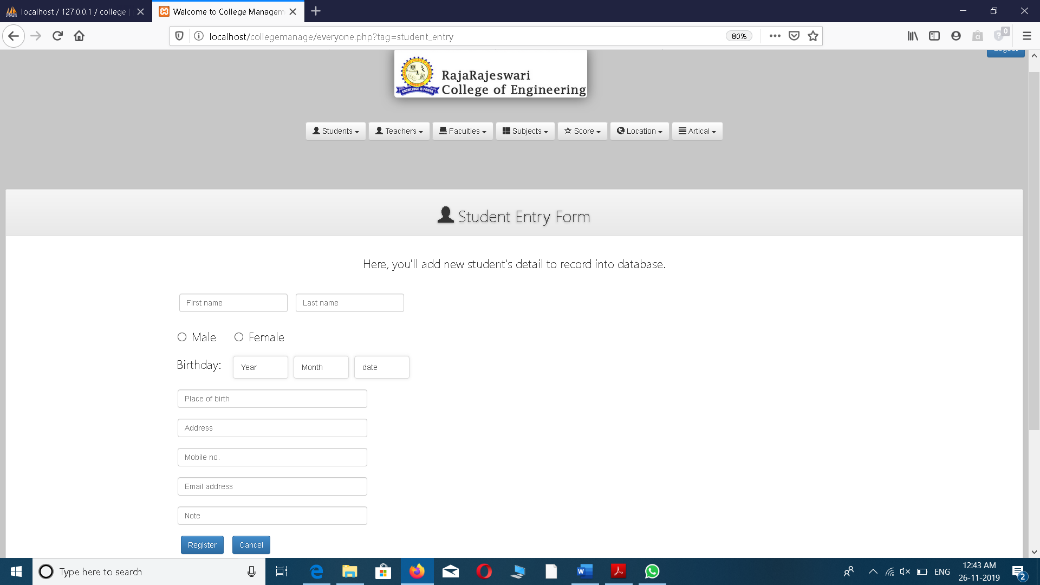
**SNAP SHOTS**



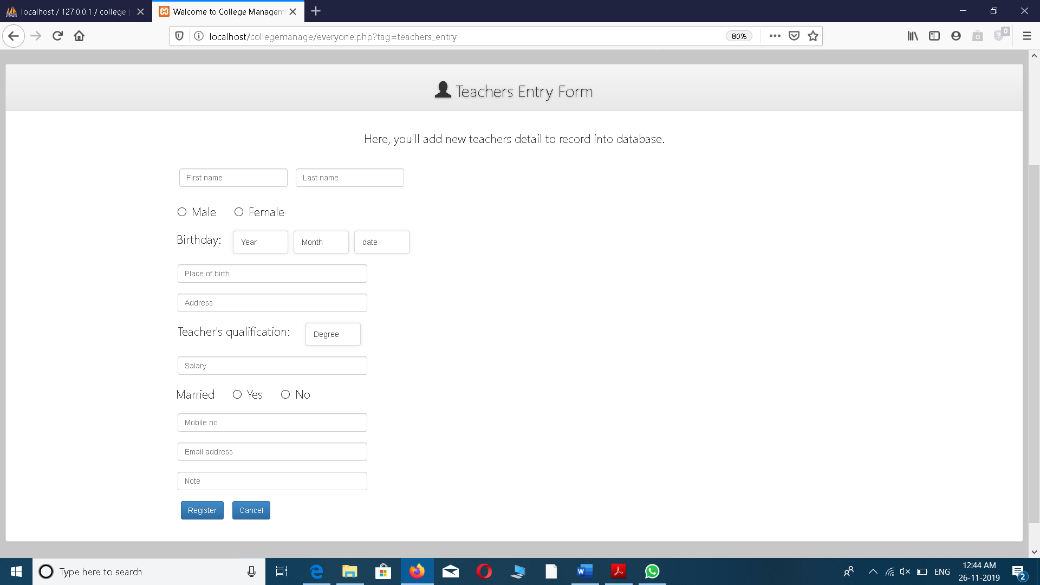
**Fig 7.1: login page**



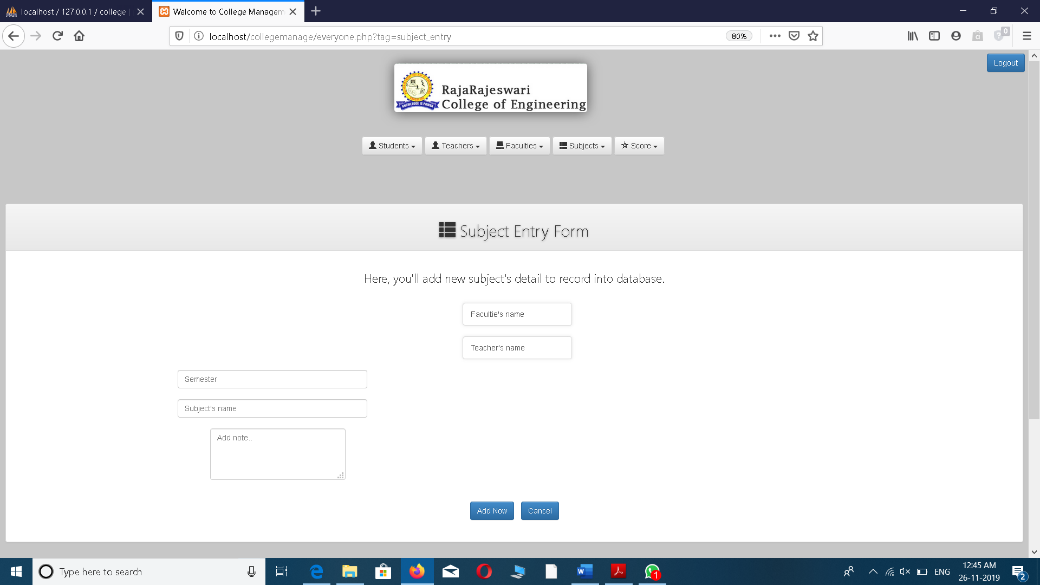
**Fig 7.2: home**



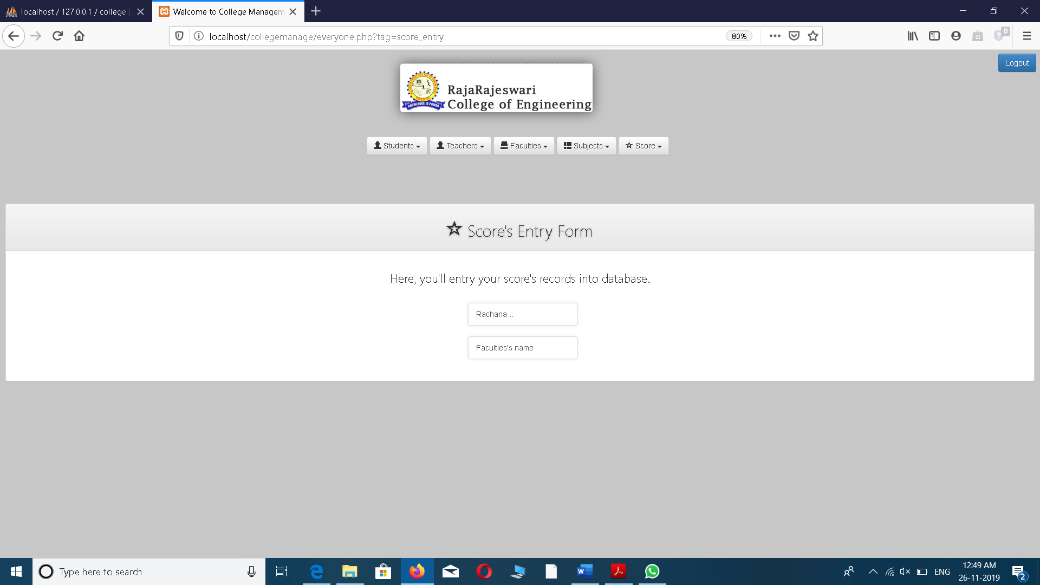
**Fig 7.3: student form**



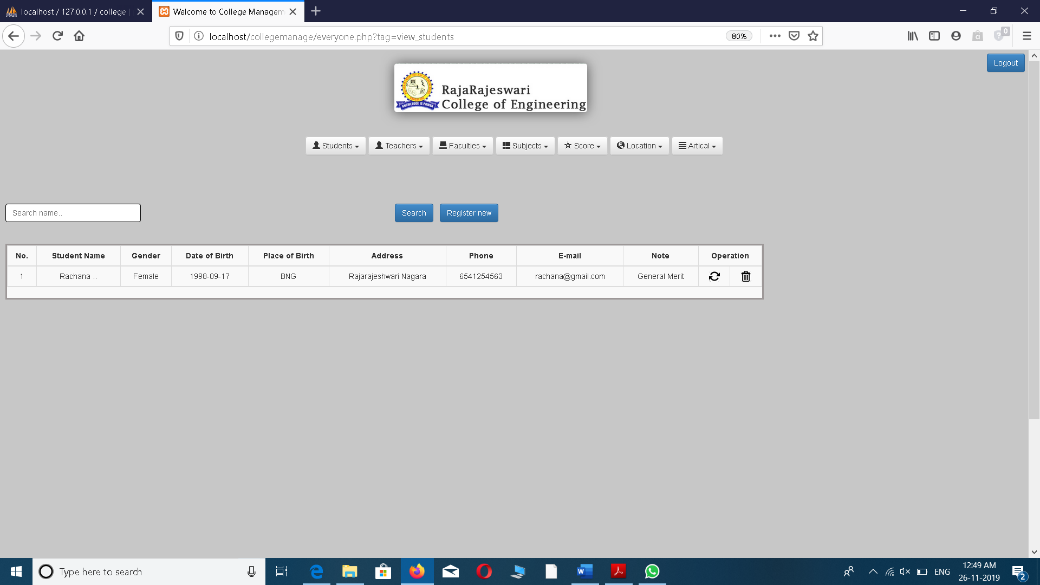
**Fig 7.4: teacher entry form**



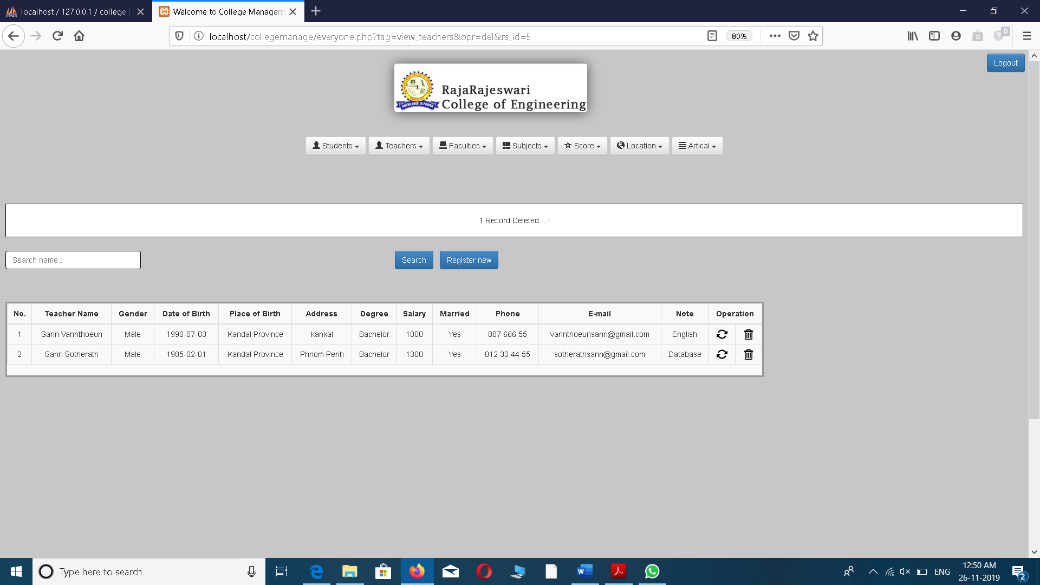
**Fig 7.5: subject form**



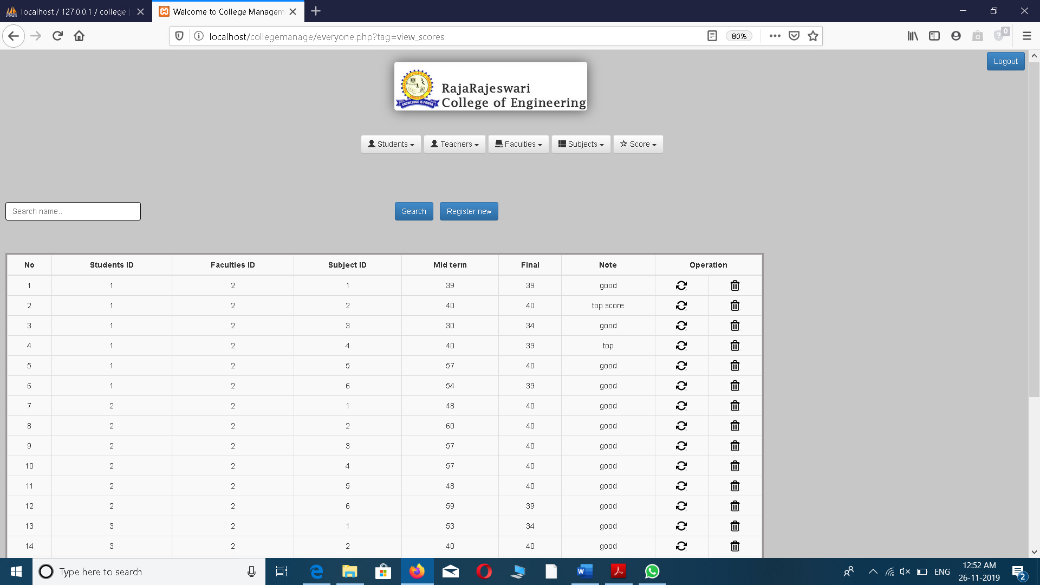
**Fig 7.6: score entry form**



**Fig 7.7: student details page**



**Fig 7.8: teacher details page**



**Fig 7.9: students marks details page**

**CONCLUSION**

The entire project has been developed and implemented as per the requirements of the user. It is found to be bug free as per the testing standard that is implemented. Any specification untraced error will be concentrated in the coming versions, which are planned for development in near future. The system was successful in depicting the aim. As the system developed was successful in depicting the operation of College Management System, it can automatically integrate with some more features. Instead of struggling with existing system with hectic procedures it is easier and stress free to use the proposed system so that user and interact and send the consignments easily and quickly.

**BIBILIOGRAPHY**

1. www.github.com
2. www.w3schools.com
3. www.getmdl.io
4. www.jquery.com
5. www.php.net
6. www.codepen.io
7. www.tutorialpoint.com