Chapter 1

INTRODUCTION

A database is a collection of logically related data, and a description of this data, designed to meet the information needs of an organization. Databases are generally separated into application areas. For example, one database may contain doctors and patient data another may contain hospital, employee data, and so on. Database is managed by a Database Management System (DBMS).

Database management system is a collection of programs that enables users to perform certain actions on database.

- Define the structure of database information (attributes, data types, constraints) storing this as meta data.
- Populate the database with appropriate information.
- Manipulate the database contents against accidental or deliberate corruption of contents.
- Share the database among multiple users, possibly concurrently.

Tuition management system is developed in a very simple and organised way which may help the institutes which are currently using some old offline methods for their administrative works. The administration maintains the central database. It contains every data about the students who join for tuition, their academic growth, subjects, classes, and employees who work for the administration.

Tuition Management System ease the work of the employee and admin. All the required information can be obtained at single place. This system will allow the admins or registrar to add, remove or edit the details of all students, teachers, subjects, classes and the fee paid by the students. Instead of using and maintaining separate book for each and every aspects of administration work, we can use this system to easily maintain the data of all the defined fields with more ease and security.

The tuition management system undertaken as a mini project is based on relevant technologies. The main aim of this mini project is to develop software for Tuition Management System. This mini project has been developed to carry out the processes easily and quickly,

which is not possible with the manual system, which are overcome by this software. This mini project is developed using PHP, HTML, CSS languages and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of workers.

The mini project analyses the system requirements and then comes up with the requirement specifications. The system is then designed in accordance with specification to satisfy the requirements. The system deals with data entry, validation confirm and updating while the interactive system deals with the interaction with administration. Thus the above features of this mini project will save transaction time and therefore increase the efficiency of the system.

1.1 Objective

These are the following objectives

- To allow only authorized user to access various function and processed available in the system.
- Add, remove and update all the details inserted before.
- To maintain the details of every aspect in the tuition.
- Reduced clerical work as most of the work done by computer.
- Provide greater speed & reduced time consumption.

1.2 Functionality

- The project is for the automation of museum management.
- The Software includes many things.
- Adding Admins and maintaining their logs.
- Storing details to database.
- Maintaining Teacher details.
- Maintaining Student and subjects details.
- Providing every details in table forms for admins.
- Class and Fee paid by students can be added.

1.3 Scope

Maintaining all the items in the tuition and managing every history and billing details is not easy thing manually. Technology is growing very fast, bringing technology into the tuition administration system makes the work easier, faster and sophisticate. Our tuition management software saves the time of organization's administration works, makes the life simple and makes the work accurate.

Chapter 2

OVERALL DESCRIPTION

2.1 Background

To survive in this modern market every organization implements so many new innovative ideas, strategies, and advanced technologies. For that they give each and every minute detail about their organization and projects to Administration. Administration works are often done by more number of employees using the typical old fashioned paper-pen method. This is always a criticizing thing even for the people standing in queue for fee payment or registration.

2.2 Problem Statement

Tuition management system plays vital role in day to day life of Tuition organization. Tuition management system mini project keeps up to date records of every data through their records. Maintaining such a big organization like tuition manually is very difficult task. In the current educational institutes, everything is done on the paper and these are highly prone to damages and requires a good amount of security and space to store. Editing and searching for specific reasons is a hectic problem in register books for administrators. So, this system reduces the struggle to maintain the records of employee to student and also for the people working in administration department.

Chapter 3

SYSTEM REQUIREMENTS

3.1 Software requirements

- Back-end tool: MYSQL version 8.0 or above
- Back-end language: Primary SQL command language.
- Front-end technologies for user interface: HTML, CSS, PHP.
- Database Connector: ODBC-MYSQL Connector version 6.7 or above.
- Operating System: Windows XAMPP or above version.

3.2 Hardware requirements

- Minimum 600MB Hard Disk space for installation
- 30GB Hard Disk space required for a typical live system with 1000-2000 events
- Recommended minimum CPU - Pentium 4, 1.7GHz
- Recommended 1GB RAM for a Central Server with 3 Nodes
- Network card

3.3 Project Requirements

- Admin can create employees
- Admin or employee can add items to database
- Should be able to add Teachers, subjects or classes.
- Should be able to issue bill
- Should be able to fetch all the reports relating to Tuition
- Should be able to manage all the records in database

3.4 User requirements

- User should be able to use computer
- User should have basic system knowledge
- He should have basic English knowledge to understand the museum system.

Chapter 4

TOOLS AND TECHNOLOGIES

4.1 XAMPP

XAMPP is the most popular PHP development environment. XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

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 the 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.

Installing XAMPP on Windows

- Download and Install Visual Studio 2008 redistributable package because XAMPP will need this to run properly.
- Download and Install XAMPP for PHP 5.X for Windows- Be sure to select the proper version of PHP as there are many choices.
- Allow the program to make changes to your machine.
- It is VERY important to install on the root of c:\. If you don't, some functions of XAMPP may have restrictions that will force you to adjust the settings in your UAC, which isn't recommended.
- Double-click on the new XAMPP Control Desktop shortcut or Use the Search feature in Windows 10 to find the XAMPP Control App, which opens up the XAMPP Control Panel Application. There you will see a column of "Start" buttons. Fig 4.1 shows the xampp control panel.

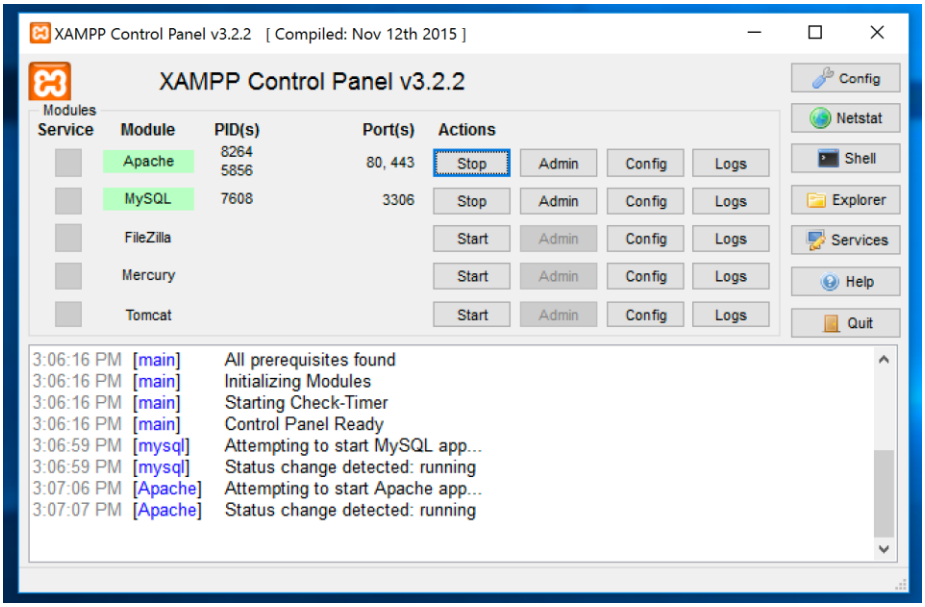

Fig 4.1 Testing your XAMPP Installation

- In your web browser, go to http://localhost. You should see the start page of XAMPP

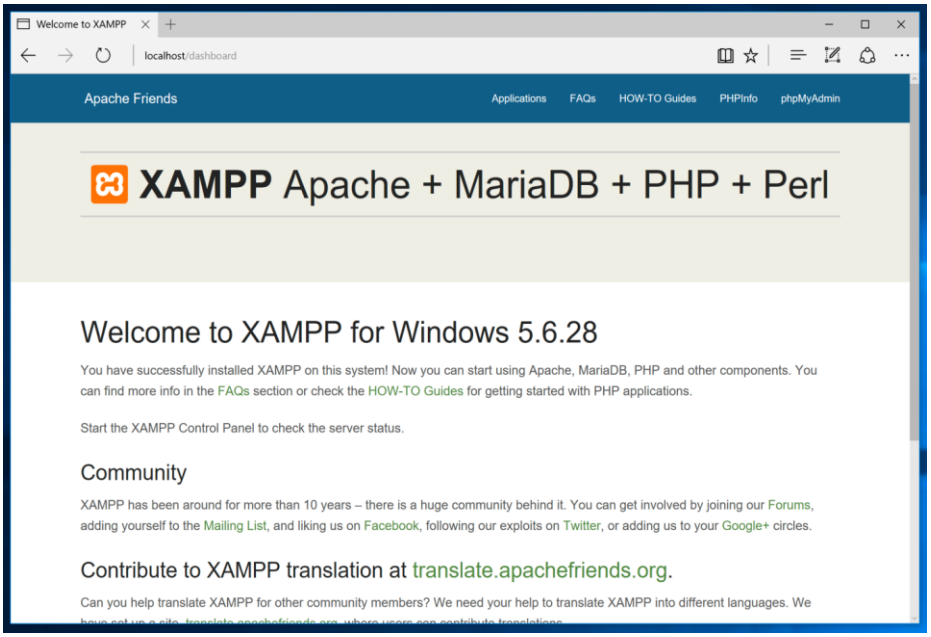

Fig 4.2 XAMPP home page

4.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL Website provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003"

refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

- **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The Main Features of MySQL

1. Internals and Portability

MySQL is written in C and C++ and tested with a broad range of different compilers. It works on many different platforms. For portability, uses in MySQL 5.5 and up. Previous series use GNU, Auto-make and Lib-tool.

MySQL Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool. It uses multi-layered server design with independent modules. Designed to be fully multi-threaded using kernel threads, to easily use multiple CPUs if they are available.

MySQL Provides transactional and nontrans actional storage engines. It uses very fast B-tree disk tables with index compression. Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.

MySQL uses a very fast thread-based memory allocation system. It executes very fast joins using an optimized nested-loop join. MySQL implements in-memory hash tables, which are used as temporary tables. It implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization. MySQL provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

2. Data Types

- Many data types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, BINARY, VARBINARY, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, and Opens spatial types. See Chapter 11, *Data Types*.

- Fixed-length and variable-length string types.

3. **Statements and Functions**

- Full operator and function support in the SELECT list and WHERE clause of queries. For example:

  MySQL> SELECT CONCAT (first name, ' ', last name)

  -> FROM citizen

  -> WHERE income/dependents > 10000 AND age > 30;

- Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions(COUNT(), AVG(), STD(), SUM(), MAX(), MIN(),and GROUP CONCAT()).

- Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax.

- Support for aliases on tables and columns as required by standard SQL.

- Support for DELETE, INSERT, REPLACE, and UPDATE to return the number of rows that were changed (affected), or to return the number of rows matched instead by setting a flag when connecting to the server.

- Support for MySQL-specific SHOW statements that retrieve information about databases, storage engines, tables, and indexes. Support for the INFORMATION_SCHEMA database, implemented according to standard SQL.

- An EXPLAIN statement to show how the optimizer resolves a query.

Testing your phpMyAdmin Installation

In your web browser, go to http://localhost/phpmyadmin. You should see the start page of phpMyAdmin as shown in fig 4.3.

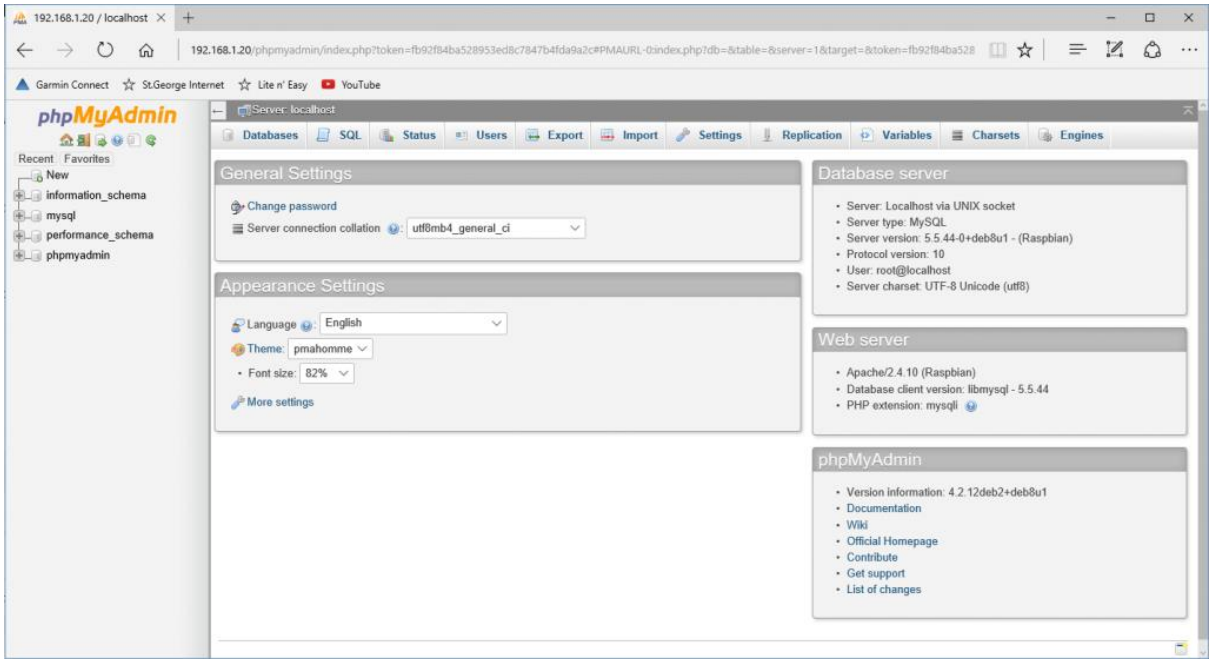

Fig 4.3 Testing phpMyAdmin Installation

Chapter 5

ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

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.

TABLES AND ATTRIBUTES

The entity relationship diagram which contains the following entities are shown in fig 5.1.

1. ADMINISTRATION (name, staff_id, phno, dept, email, password);
2. TEACHER (teach_id, name, dob, gender, email, staff_id);
3. STUDENT (sid, sname, place, gender, dob, email, tch_id);
4. CLASS (class_no, day, from, to, teach_id);
5. SUBJECT (sub_code, sub_name, cls_no, tch_id);
6. FEE (bill_no, total_pay, date, sid);

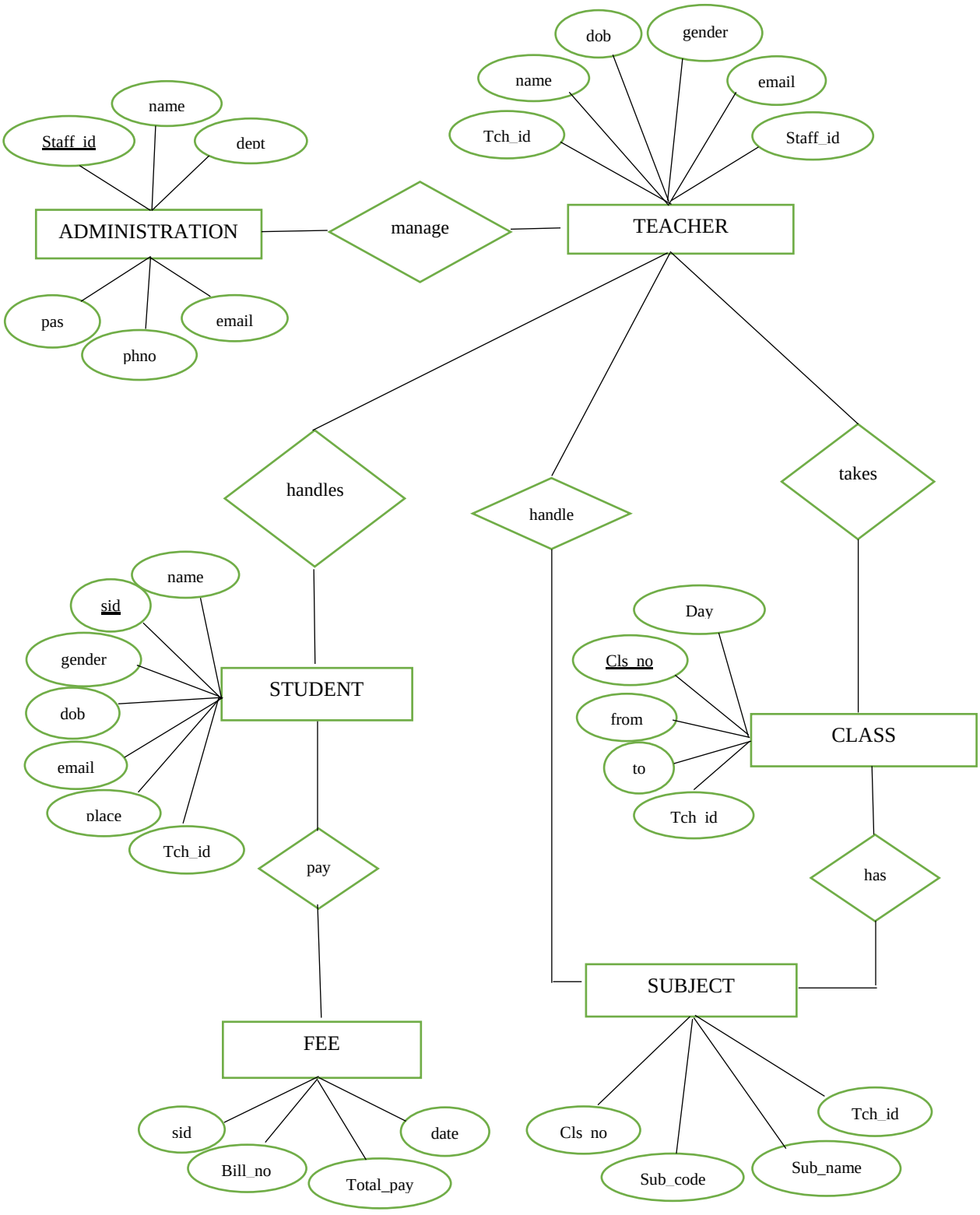

Fig 5.1: Tuition management E-R diagram.

Chapter 6

DATA TABLES

- **Table Name :-** ADMIN DETAILS.
- **Description :-** This table shows the attributes of Admins.
- **Primary key :-** staffid

Table 6.1: Admin Details

Field	Type	Null	Key	Default	Extra
name	varchar(40)	YES		Null	
staffid	Int(11)	NO	PRI	Null	
Department	varchar(40)	YES		Null	
Phone	Double	YES		Null	
Email	varchar(40)	YES		Null	
password	varchar(40)	YES		Null	

- **Table Name :-** TEACHER DETAILS.
- **Description :-** This table shows the attributes of Teachers.
- **Foreign key :-** staffId

Table 6.2: Teacher Details

Field	Type	Null	Key	Default	Extra
name	varchar(40)	YES		Null	
teacherId	varchar(40)	NO	PRI	Null	
Email	varchar(40)	YES		Null	
Gender	varchar(40)	YES		Null	
DOB	varchar(40)	YES		Null	
staffId	Int(11)	YES	MUL	Null	

- **Table Name :-** STUDENT DETAILS.
- **Description :-** This table shows the attributes of Students.
- **Foreign Key :-** teacherId

Table 6.3: Student Details

Field	Type	Null	Key	Default	Extra
name	varchar(40)	YES		Null	
studentId	varchar(40)	NO	PRI	Null	
Email	varchar(40)	YES		Null	
Gender	varchar(40)	YES		Null	
DOB	varchar(40)	YES		Null	
Place	varchar(40)	YES		Null	
teacherId	varchar(40)	YES	MUL	Null	

- **Table Name :-** CLASS DETAILS.
- **Description :-** This table shows the attributes of Class.
- **Foreign Key :-** teacherId

Table 6.4: Class Details

Field	Type	Null	Key	Default	Extra
classNo	varchar(40)	NO	PRI	Null	
Day	varchar(40)	YES		Null	
From	varchar(40)	YES		Null	
To	varchar(40)	YES		Null	
teacherId	varchar(40)	YES	MUL	Null	

- **Table Name :-** SUBJECT DETAILS.
- **Description :-** This table shows the attributes of Subject.
- **Foreign Key :-** classNo, teacherId

Table 6.5: Subject Details

Field	Type	Null	Key	Default	Extra
subjectName	varchar(40)	YES		Null	
subCode	Int(11)	NO	PRI	Null	
classNo	varchar(40)	YES	MUL	Null	
teacherId	varchar(40)	YES	MUL	Null	

- **Table Name :-** FEE DETAILS.
- **Description :-** This table shows the attributes of Fee.
- **Foreign Key :-** studentId

Table 6.6: Fee Details

Field	Type	Null	Key	Default	Extra
studentId	varchar(40)	YES	MUL	Null	
billNo	varchar(40)	NO	PRI	Null	
totalPay	varchar(40)	YES		Null	
date	varchar(40)	YES		Null	

Chapter 7

SNAPSHOTS

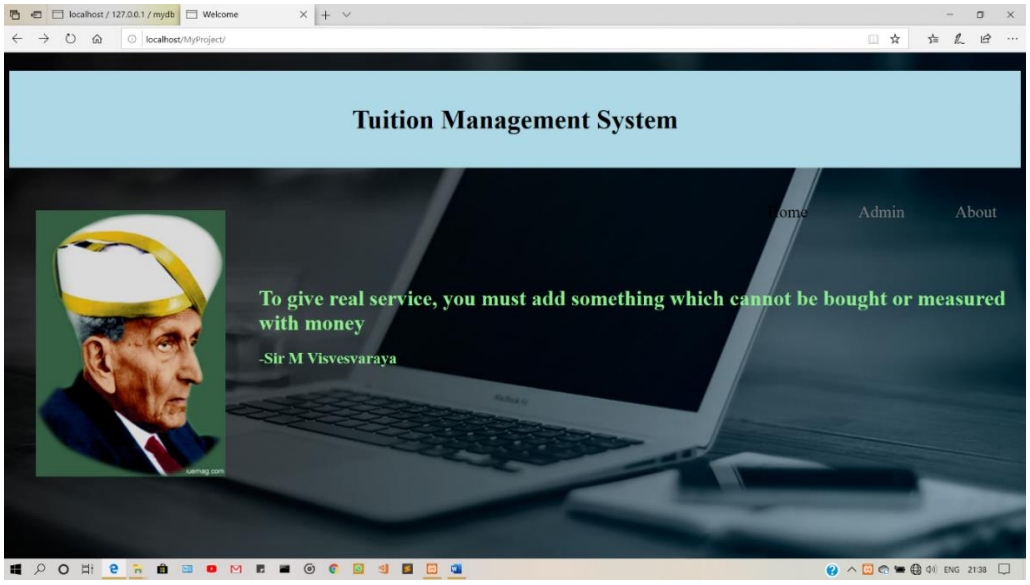

Fig 7.1: Home page

- Fig 7.1 shows the home page of the project

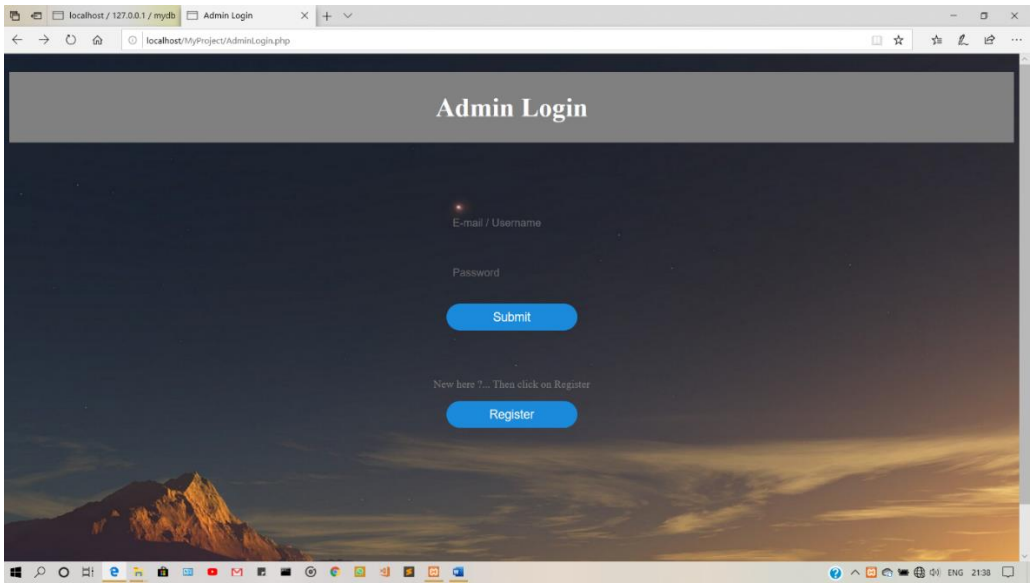

Fig 7.2: Admin Login

- Fig 7.2 shows the Login page of the Admin and Register options for new admins

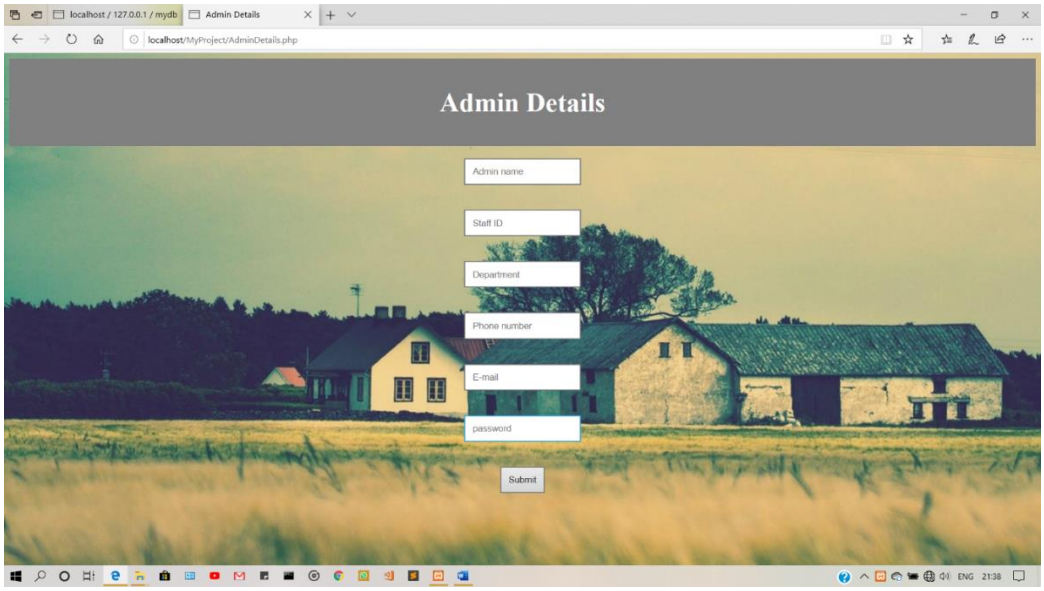

Fig 7.3: Admin Register

- Fig 7.3 gives the Admin register details to fill

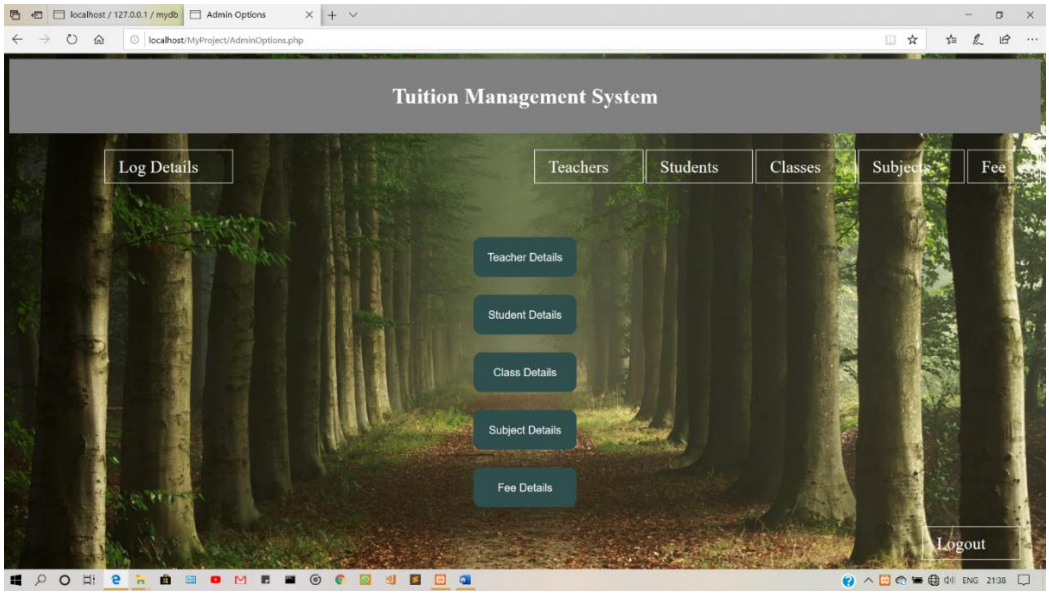

Fig 7.4: Admin Options

- Fig 7.4 shows the first page after login and every options that Admin can do

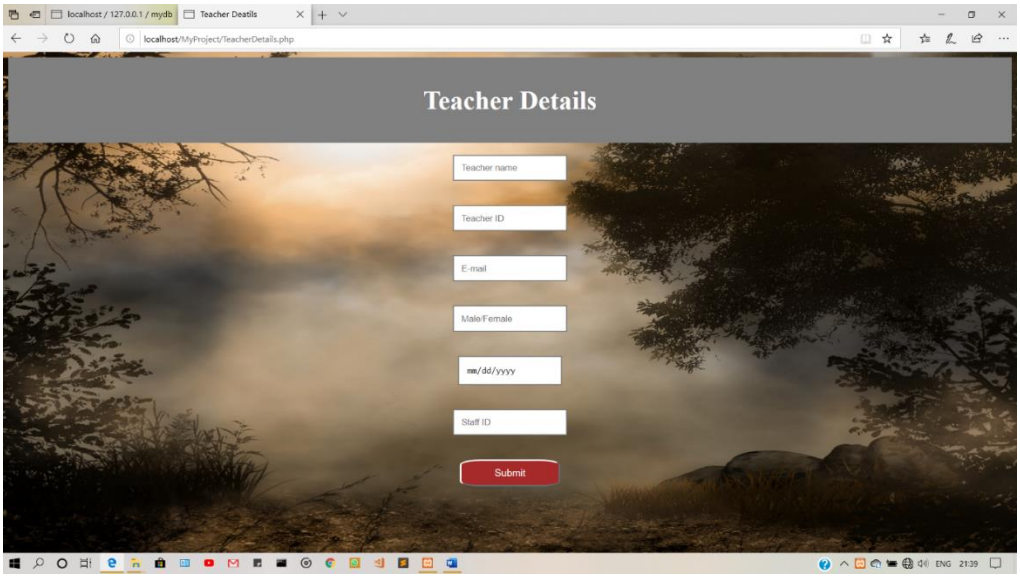

Fig 7.5: Insert Teacher

- Fig 7.5 shows the insertion of new teacher to the database

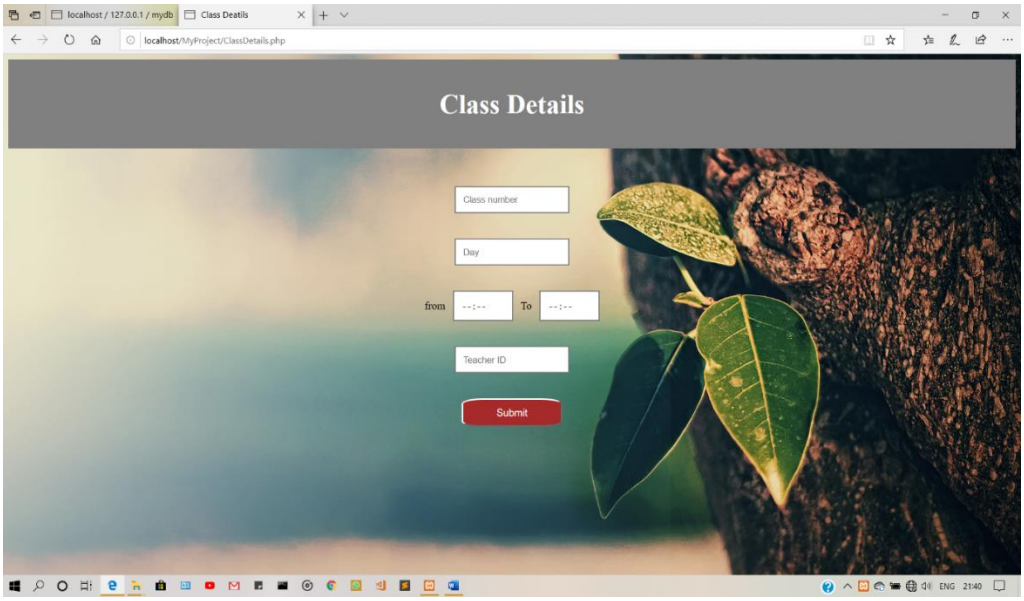

Fig 7.6: Insert Class

- Fig 7.6 shows the insertion of new class to the database

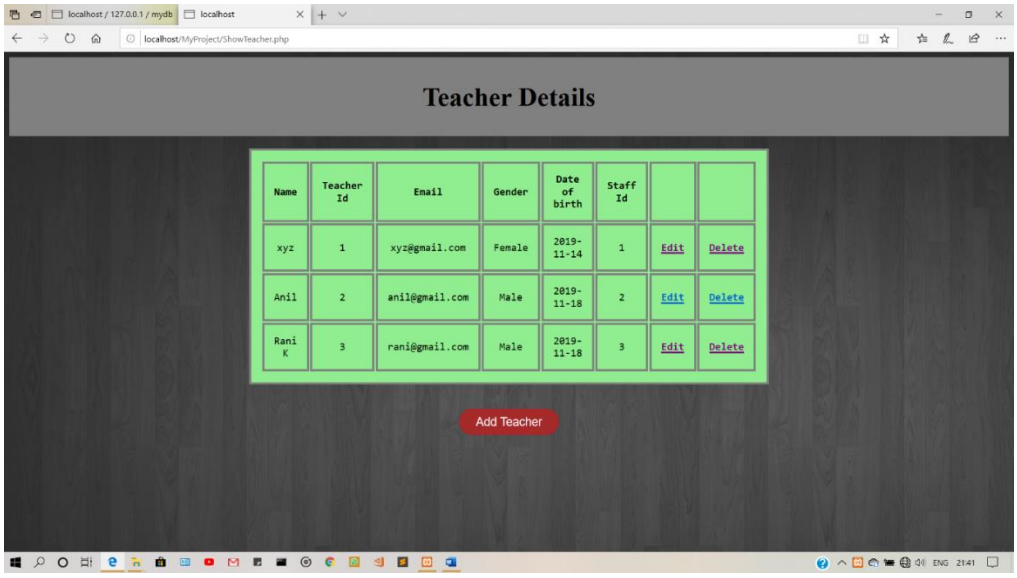

Fig 7.7: Teacher Details

- Fig 7.7 shows every details of the teacher with edit and delete options.

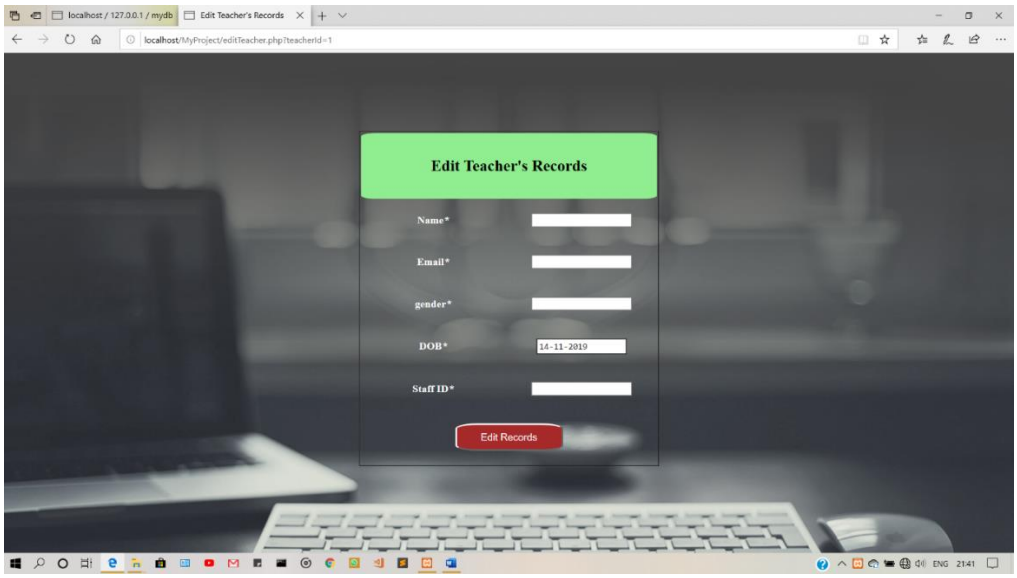

Fig 7.8: Edit Teacher Details

- Fig 7.8 shows the edit page of teacher details.

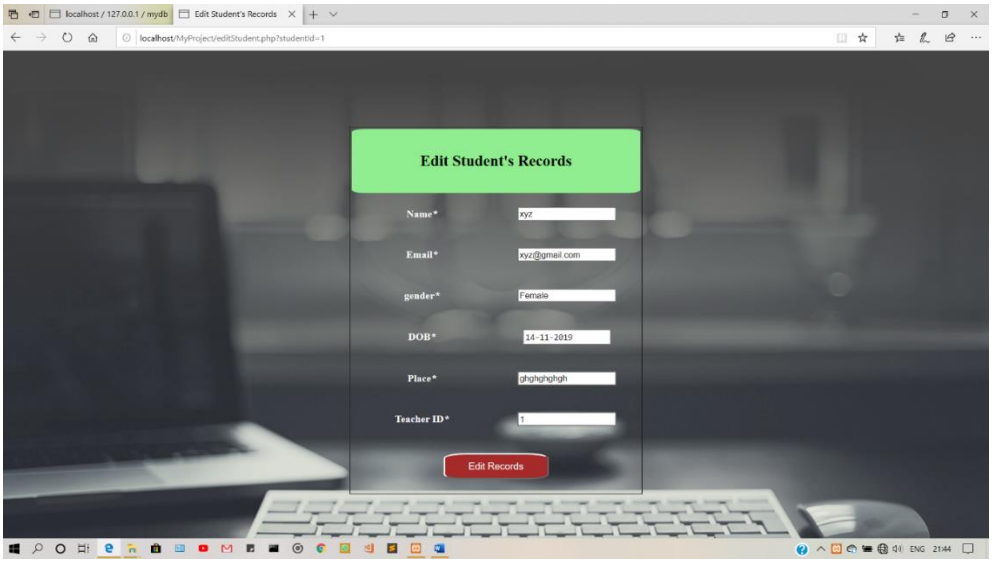

Fig 7.9: Edit Students Details

- Fig 7.9 shows the edit page of student details.

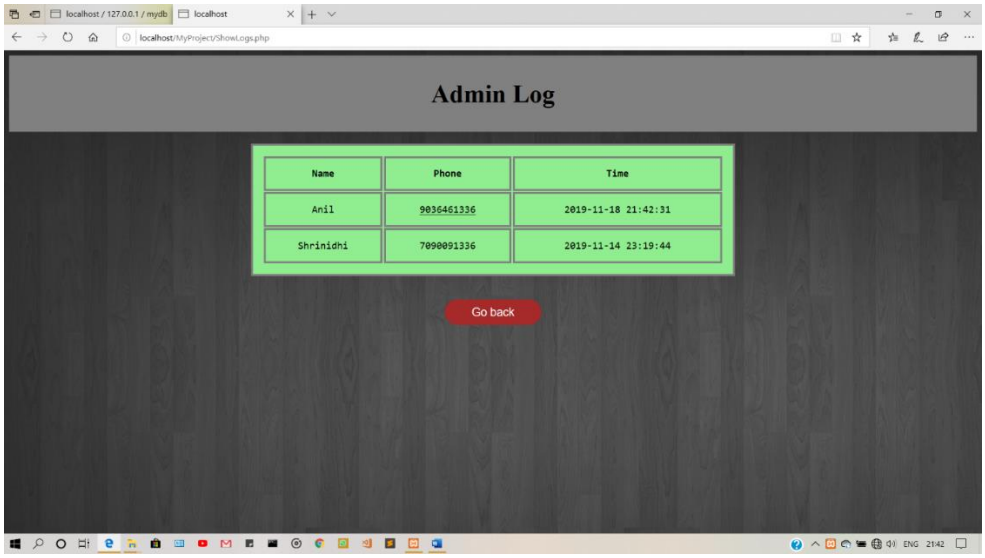

Fig 7.10: Admin Logs

- Fig 7.10 shows the log page which contains the details of admins whenever they register.