**1. INTRODUCTION**

**1.1PROJECT OVERVIEW**

School Management System is a complete software solution for managing the activities of a School. This system helps the school authorities for better interaction between students, teachers, parents . The main purpose of building this system is that, the parents of students are very busy now days, so they can’t monitoring their children and their activities properly and regularly. This system will reduce the manual operation required to handle student, teacher parent registration and teachers allocations etc.

**1.2 PROJECT SPECIFICATION**

Currently the school is dealing manual records and it is too difficult for them to pick up a particular record for reference. Every detail is entered in books of the school. This project is aimed towards recording a considerable number of student records and needs online assistance for managing records of students. This website is user-friendly, ‘quick to learn’ and reliable for the above purpose. This system is intended to be a stand-alone product andshould not depend on the availability of other website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

The system includes 4 main modules. They are:

**ADMIN:**

The admin is the overall controller of the system.

The admin can add and delete users, manages the system.

Admin add other information such as insert religion, community for the purpose of registration

**TEACHERS:**

They can view and edit their profile. Also Manage exam mark and assignment mark.

**STUDENTS**

They can view and edit their profile and view their marks.

**PARENTS:**

They can view and edit their profile and also can view all academic details of their students.

**2. SYSTEM STUDY**

**2.1 INTRODUCTION**

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute’s detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

**2.2 PRODUCT PERSPECTIVE**

The **SMS.Com** is an independent product that automates some of the tasks that are carried out in **Surya Jyothi Public School,Kanamala**which is now carried out in paper works. It mainly focuses on students and their academic performance status. This system keeps all the details of the students and basic details about staffs. It also handles exam scheduling and mark entering. The project SMS.Comis aimed towards recording a considerable number of student records and needs online assistance for managing records of students. Website is developed user-friendly, quick to learn and reliable for the above purpose. SMS.Com is a stand-alone product andshould not depend on the availability of other website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

**2.3 EXISTING SYSTEM**

Currently the school is dealing manual records and it is too difficult for them to pick up a particular record for reference. Every detail is entered in books of the school.One of the major drawback of existing system is thatIt is less user-friendly and take a lot of time to access the records. And also the paper record doesn’t last permanently and can damage due to long time stack.

**2.4 PROPOSED SYSTEM**

The proposed system will handle all the difficulties of the existing systems. The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system.The software system should be convenient for its users like students, teachers, and parents. The aim of proposed system is to develop a system with improved facilities.It is designed to provide an enhanced interaction between students, teachers, parents.

**2.5 USER CHARACTERISTICS:**

School Management System has 4 users.

1. **Admin Module:**

The admin can access the entire system functionalities. The Admin can login to the system, can register new staffs, parents,new students according to the class, add and delete other information such as insert religion, community for the purpose of registration. Allocation of class teacher and subject teacher is carried out by the admin.Admin can add new stream, if new stream get started in the school and add corresponding subjects for study.

1. **Teacher**

They can view and edit their profile. View their students. Schedule exams and enter mark for each exam and create assignments for students.

**3**. **Student**

Students can view and edit their profile. They can see all exam mark They can change their passwords

1. **Parent**

They can view and edit their profile. They can see their students details. And also change their passwords

.

**3.SYSTEM ANALYSIS**

**3.1 FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

The following are its features:

### **3.1.1 Economical Feasibility**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

All the resources are already available, it give an indication of the system is economically possible for development. The developed system works with in the budget. SMS.Com will be a simple platform for the school authority to maintain student’s record for future references and manage students’ academic details. It is less free of cost. Using this system entire staff and students can register with less free of cost.

**3.1.2 Technical Feasibility**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

Technical guarantees of data security is provided by encrypting the passwords using md5 algorithm,

The project should be developed such that the necessary functions and performance are achieved within the constraints Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using the PHP in front end and MySql server in back end, the project is technically feasible for development.

**3.1.3 Behavioral Feasibility**

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.GUI is less complicate to handle so that users can handle it without any confusion. Even though it is less complicated to handle the software, initial familiarisation and primary training is needed for them.

**3.2 SYSTEM SPECIFICATION**

**3.2.1 Hardware Specification**

Processor - Pentium 4/AMD Dual Core and above

RAM - 4 GB

Hard disk - 500 GB

**3.2.2 Software Specification**

Front End - PHP

Backend –My-SQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

**3.3 SOFTWARE DESCRIPTION**

### **3.3.1 PHP**

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym .PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

**3.3.2 MySQL**

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

**• 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. 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.

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

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 backend, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

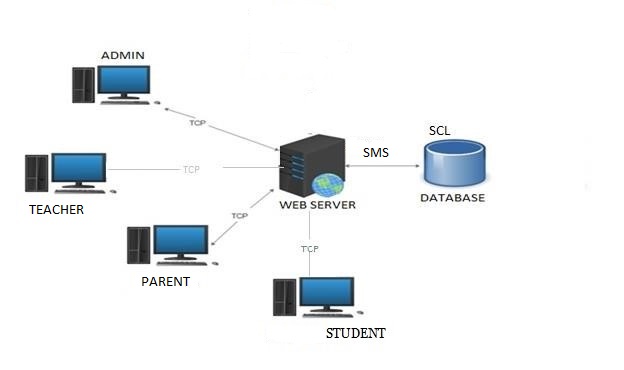
• **A large amount of contributed MySQL software is available**.

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.

**4.SYSTEM DESIGN**

**4.1 ARCHITECTURAL DESIGN**

This section describes the components of **SMS**



**Figure 1: Architectural Design**

The admin, teacher, parent,student can accesses the SMS through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System’s application program processes the user’s request and provides the required services by taking data from the system database.

4.2 **MODULE DESIGN**

**Admin Module**

The admin is the overall controller of the system. The admin can add, delete and edit users, manages the system. Admin add other information such as insert class, division and religion, for the purpose of registration.

|  |  |
| --- | --- |
| Register new student, teacher, parent | Allocation of class-teacher -subject |
| Create new class and division | Allocate students to class |
| Add religion, class, division, subject | View all students |
| View all staff | View all parents |

**Teacher Module**

They can view and edit their profile. Schedule exams and enter mark and create assignments for students. They can also change their passwords.

|  |  |
| --- | --- |
| View/Edit profile | View students |
| Create exam | Enter marks |
| Create assignments | Change Password |

**Student Module**

Students can view/edit their profile. They can see exam mark and Also change their passwords

|  |  |
| --- | --- |
| View profile | View exams mark |
| Edit profile | Change Password |

**Parent Module**

Parents can view/edit their profile. They can see exam mark of their student and Also change their passwords

|  |  |
| --- | --- |
| View profile | View exams mark |
| Edit profile | Change Password |

**4.3 DATA FLOW DIAGRAMS**

Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It’s a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process-oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

**Main symbols used in the data flow diagram are:**

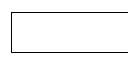
1. Circle represents a process that transforms incoming data flows in to outgoing data flows.



1. A square defines a source and destination of system data.



1. Arrow identifies data in motion.
2. An open rectangle defines a data store, data at rest or temporary repository of data.



.

**Steps to Construct Data Flow Diagrams:-**

Four steps are commonly used to construct a DFD:

 Process should be named and numbered for easy reference. Each name should be representative of the process.

 The destination of flow is from top to bottom and from left to right.

 When a process is exploded in to lower level details they are numbered.

 The names of data stores, sources and destinations are written in capital letters.

**Rules for constructing a Data Flow Diagram**

 Arrows should not cross each other.

 Squares, circles and files must bear names.

 Decomposed data flow squares and circles can have same names.

 Draw all data flow around the outside of the diagram.

**Data Flow Diagrams for School Management System**

**Context Diagram**

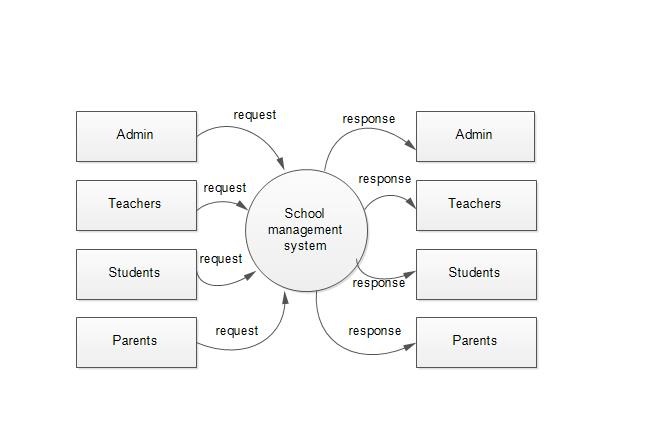
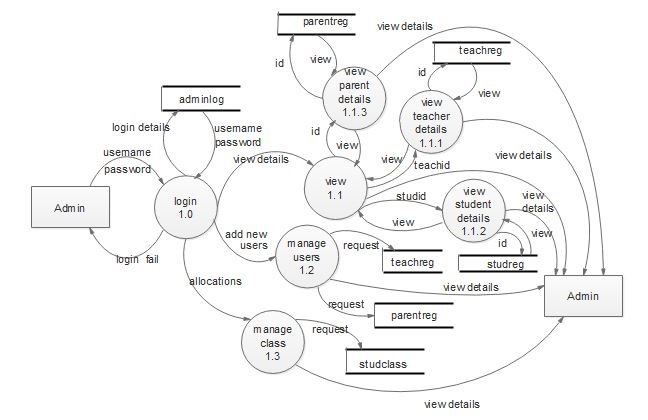


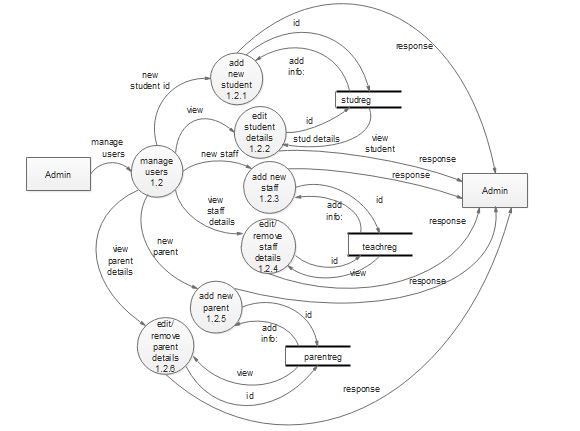
Figure 2: context diagram of SMS

**Level 1 DFD for ADMIN**

****

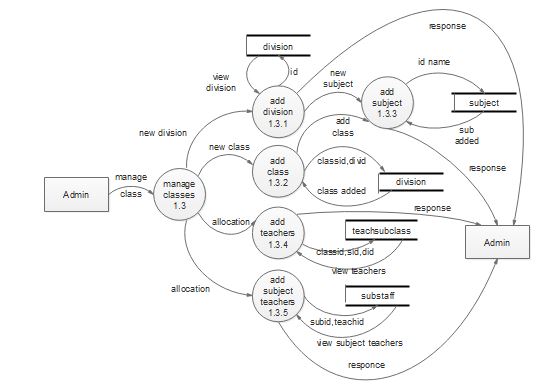
**Figure 3: Level 1 data flow diagram for Admin**

**Level 2 DFD for ADMIN**



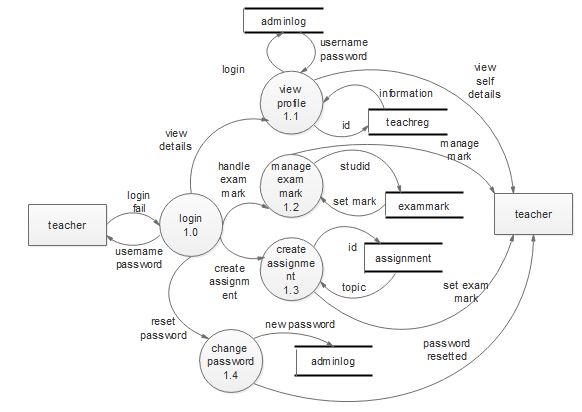
**Figure 4: Level 2 data flow diagram for Admin**

**Level 2 DFD for ADMIN**



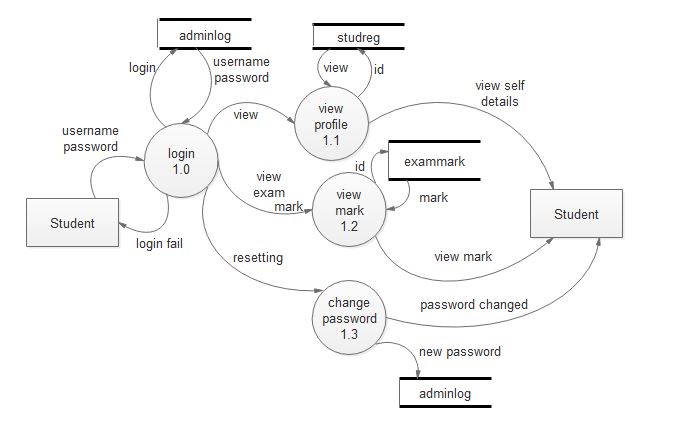
**Figure 5: Level 2 data flow diagram for Admin**

**Level 1 for Teacher**



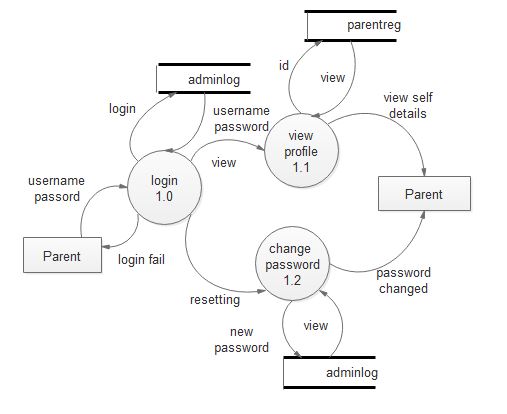
**Figure 6: Level 1 data flow diagram for Teacher**

**Level 1 for Student**



**Figure 7: Level 1 data flow diagram for Student**

**Level 1 for Parent**

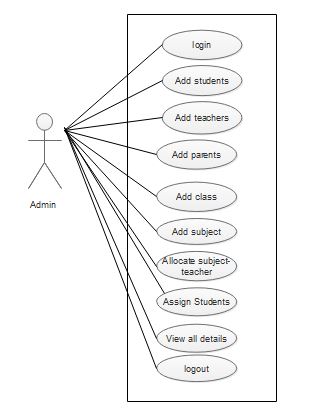


**Figure 8: Level 1 data flow diagram for a Parent**

**4.4 UML DIAGRAM**

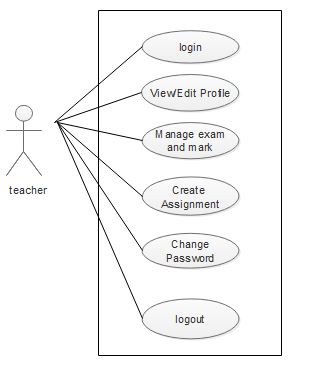
**4.4.1 USECASE DIAGRAM**

**USE CASE DIAGRAM – ADMIN**

****

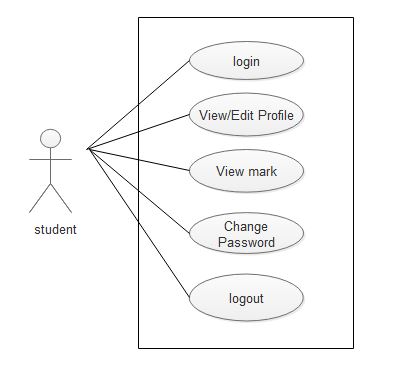
**Figure 9: use case diagram for admin**

**USE CASE FOR TEACHER**

****

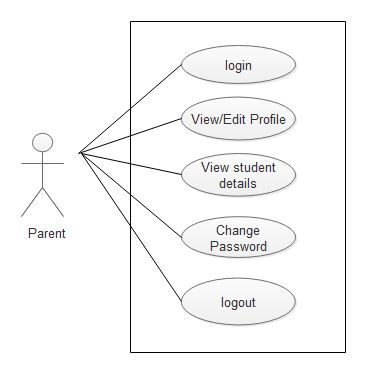
**Figure 10: use case diagram for teacher**

**USE CASE FOR STUDENT**

****

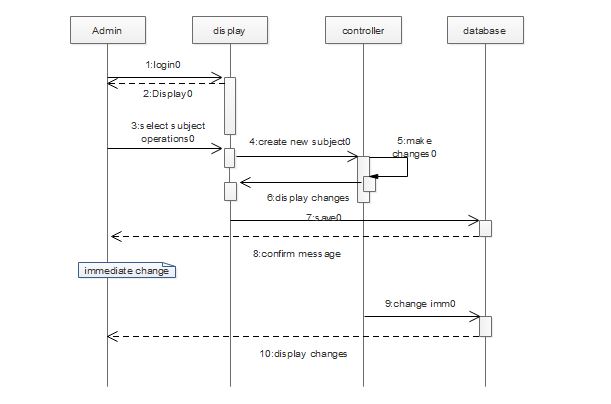
**Figure 11: use case diagram for student**

**USE CASE FOR PARENT**

****

**Figure 12: use case diagram for parent**

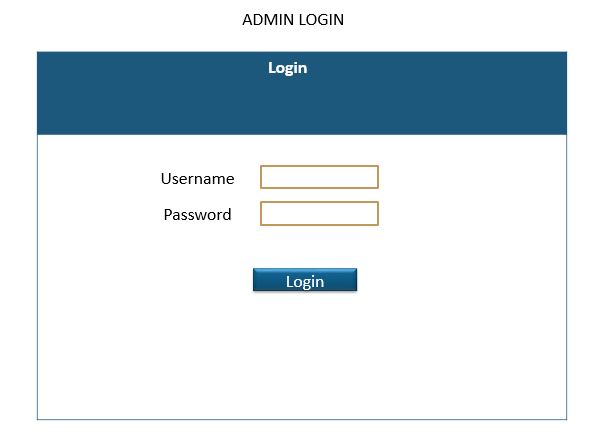
**4.4.2 SEQUENCE DIAGRAM**



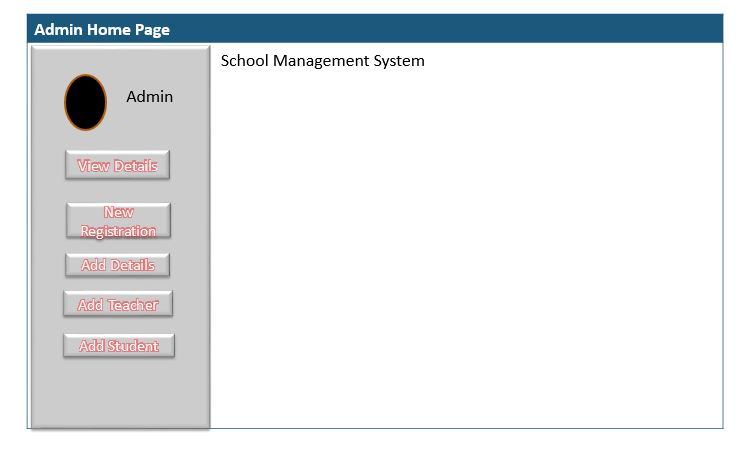
**Figure 13 : Sequence Diagram**

**4.5 USER INTERFACE DESIGN**

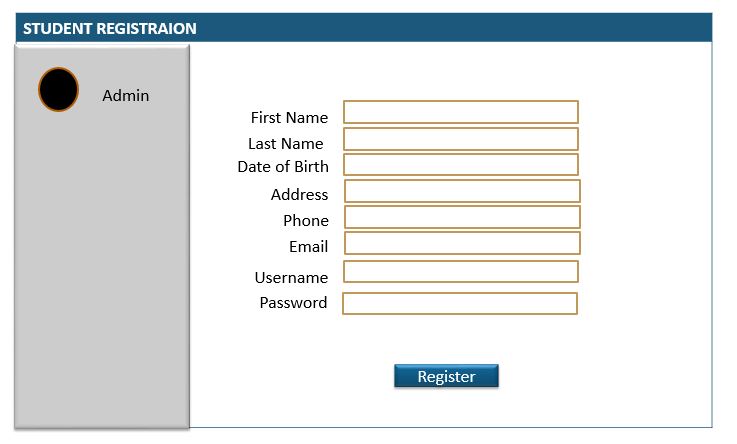
**4.5.1 USER INTERFACE DESIGN**



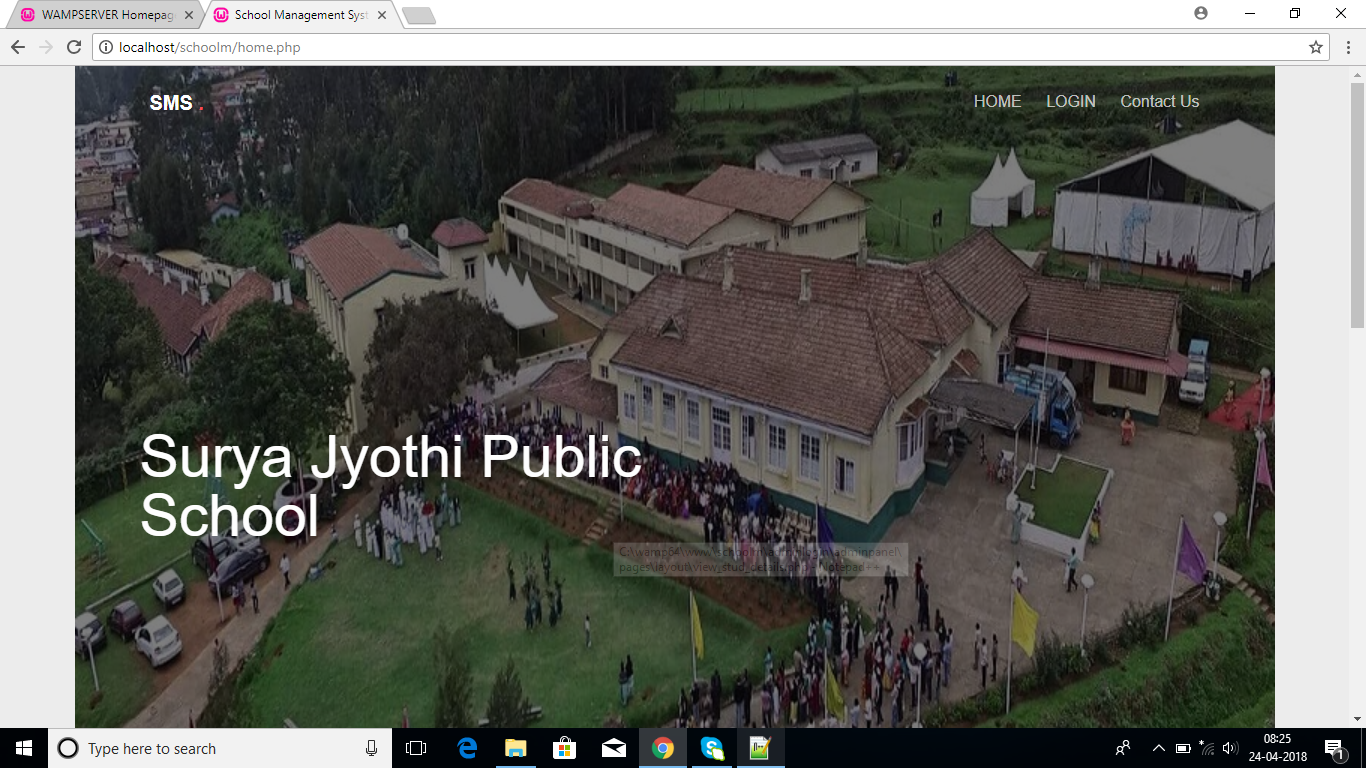
**ADMIN DASHBOARD**

****

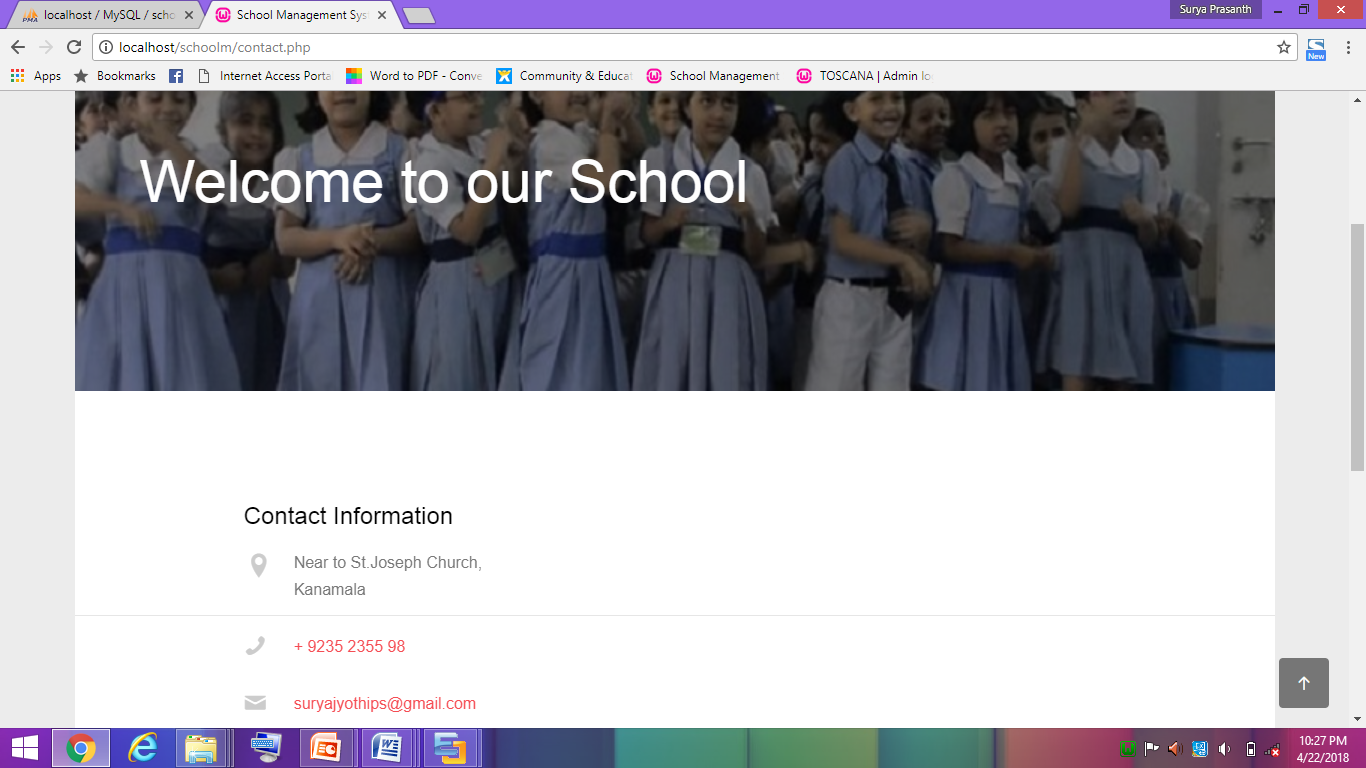
**USER REGISTRATION**

****

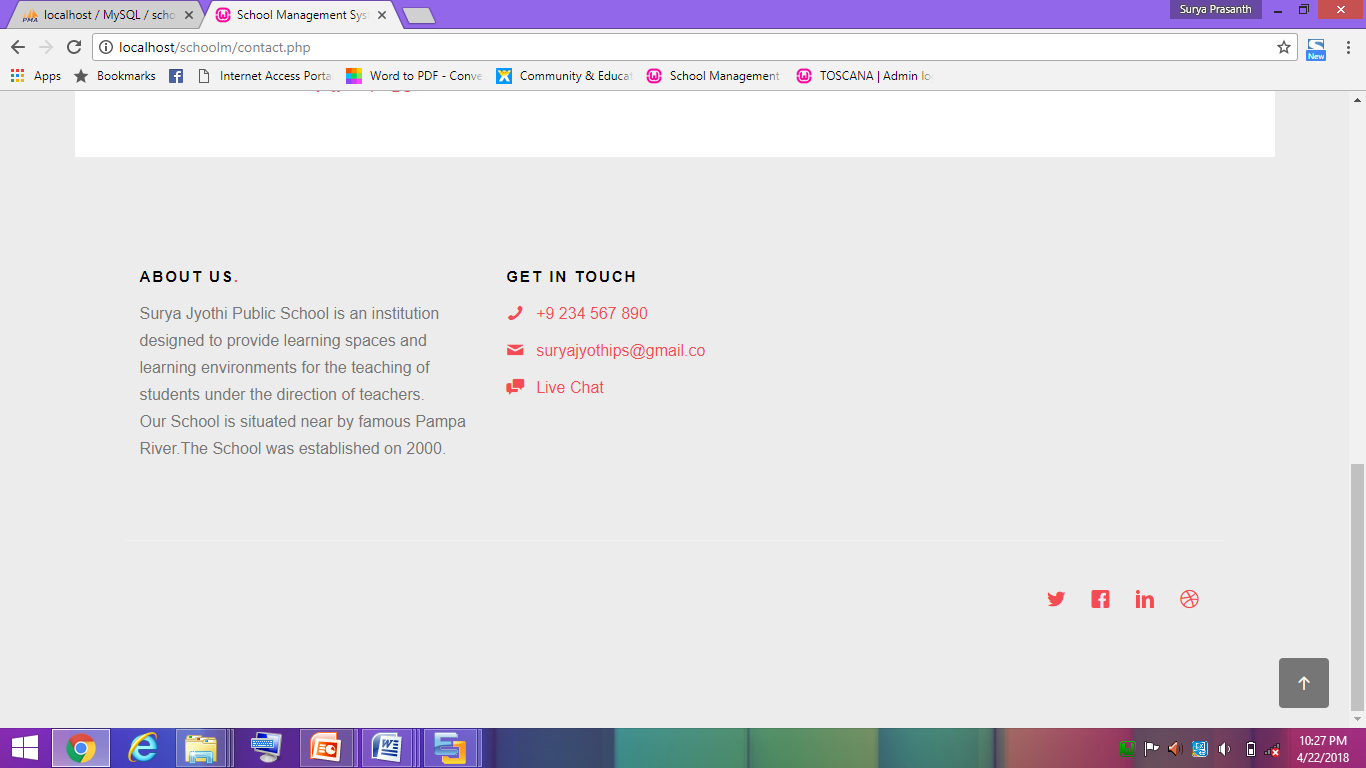
**4.6 SCREEN SHORTS**

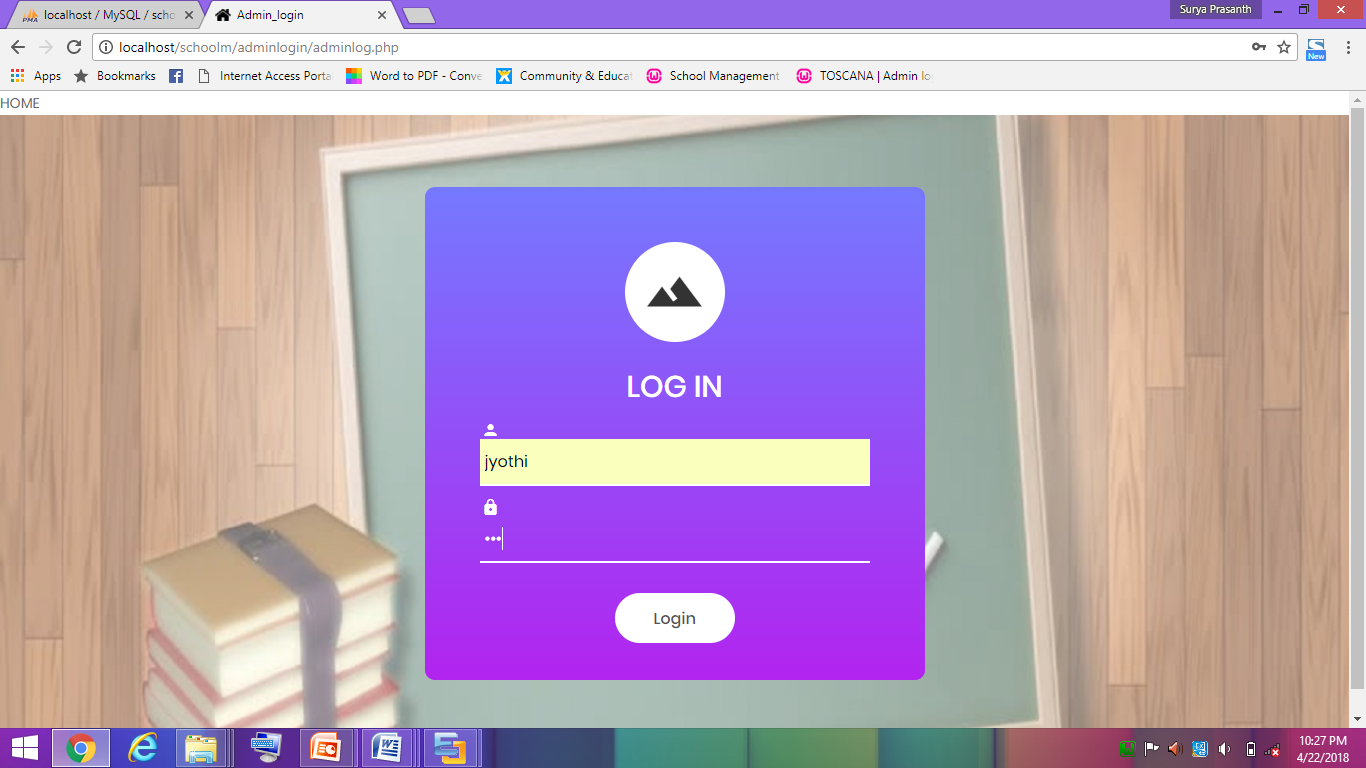


**Figure 14: Home Page**

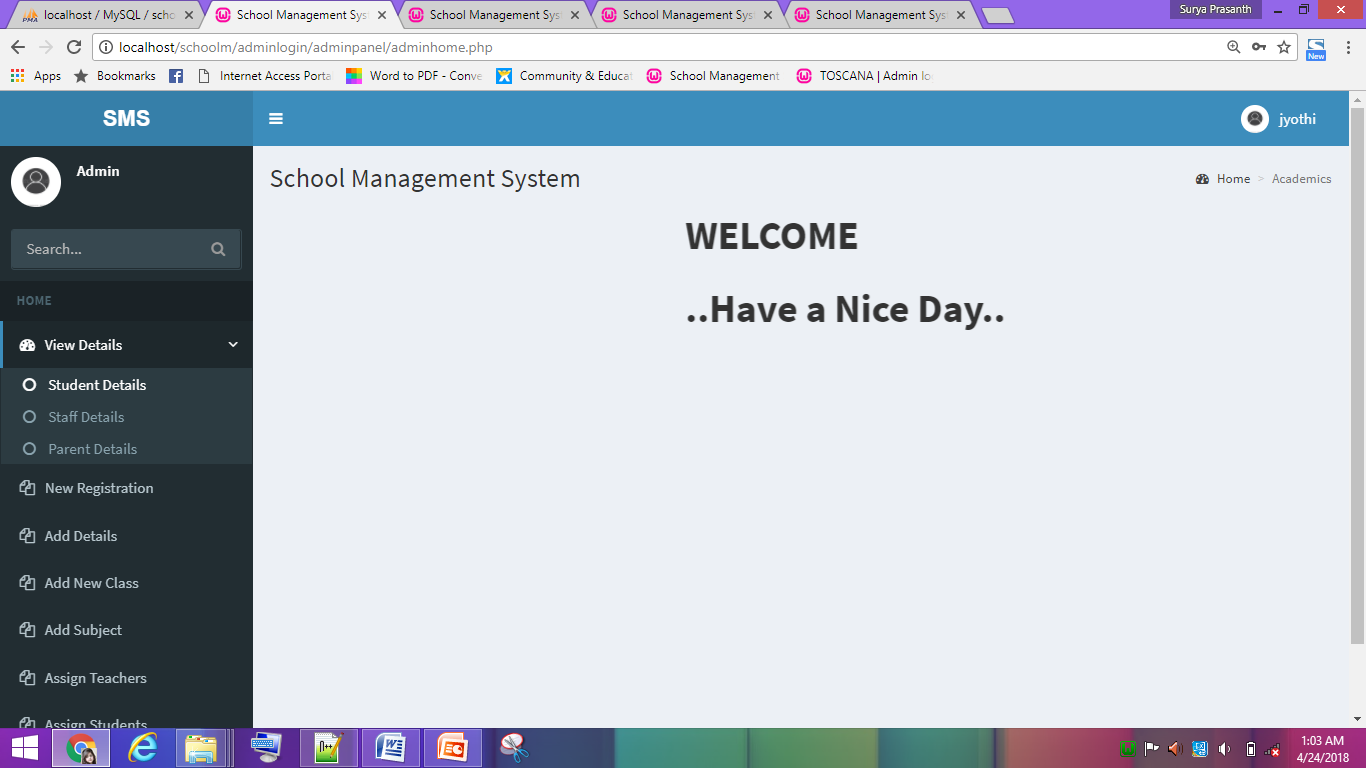


**Figure 15: Contact Details**

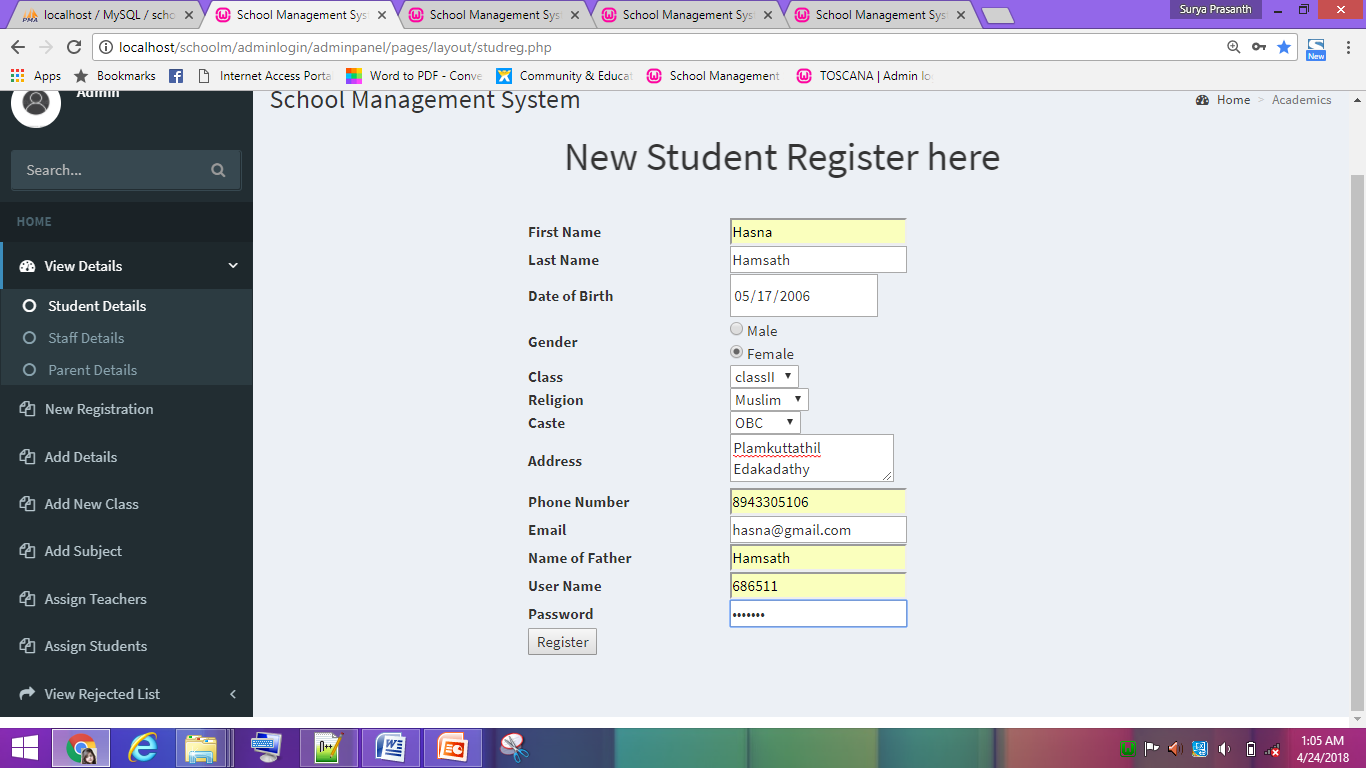
  
 **Figure 16: About page**

****

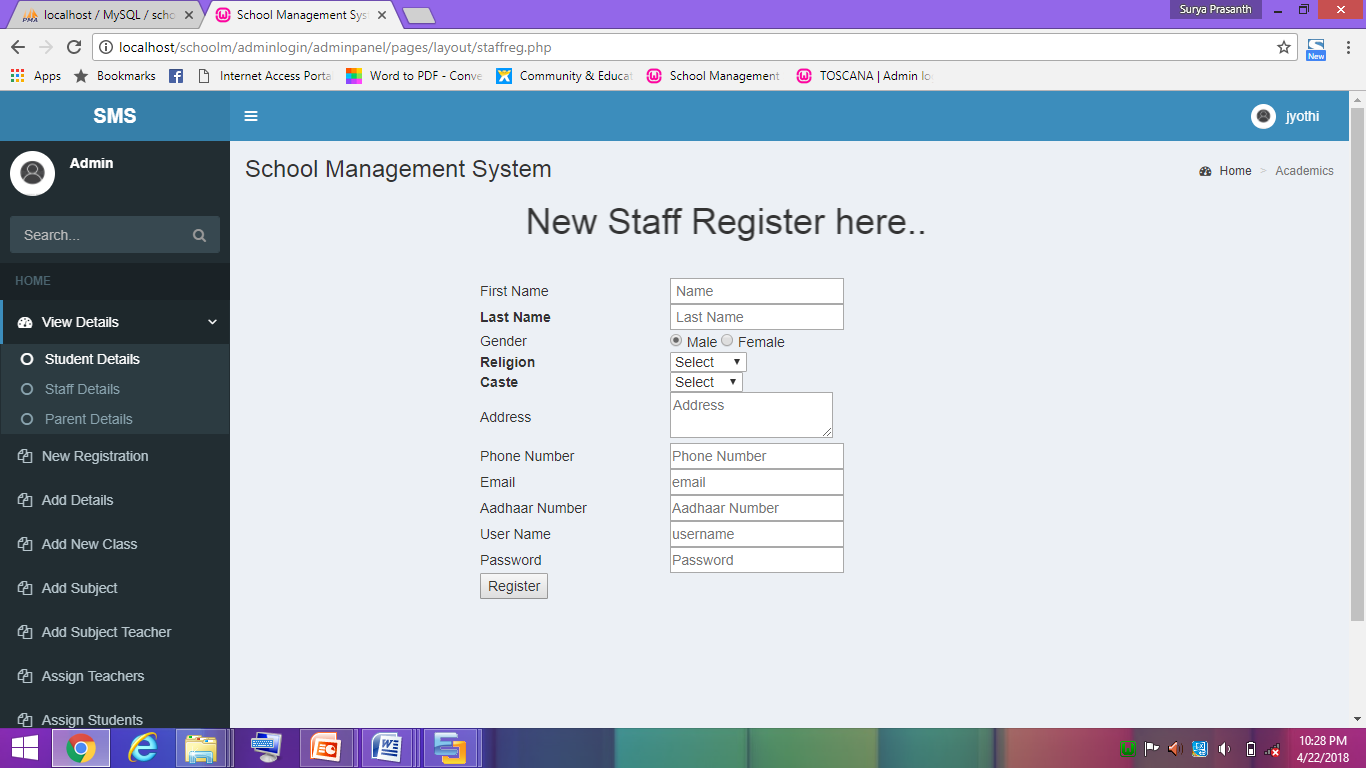
**Figure 17: Admin login page**

****

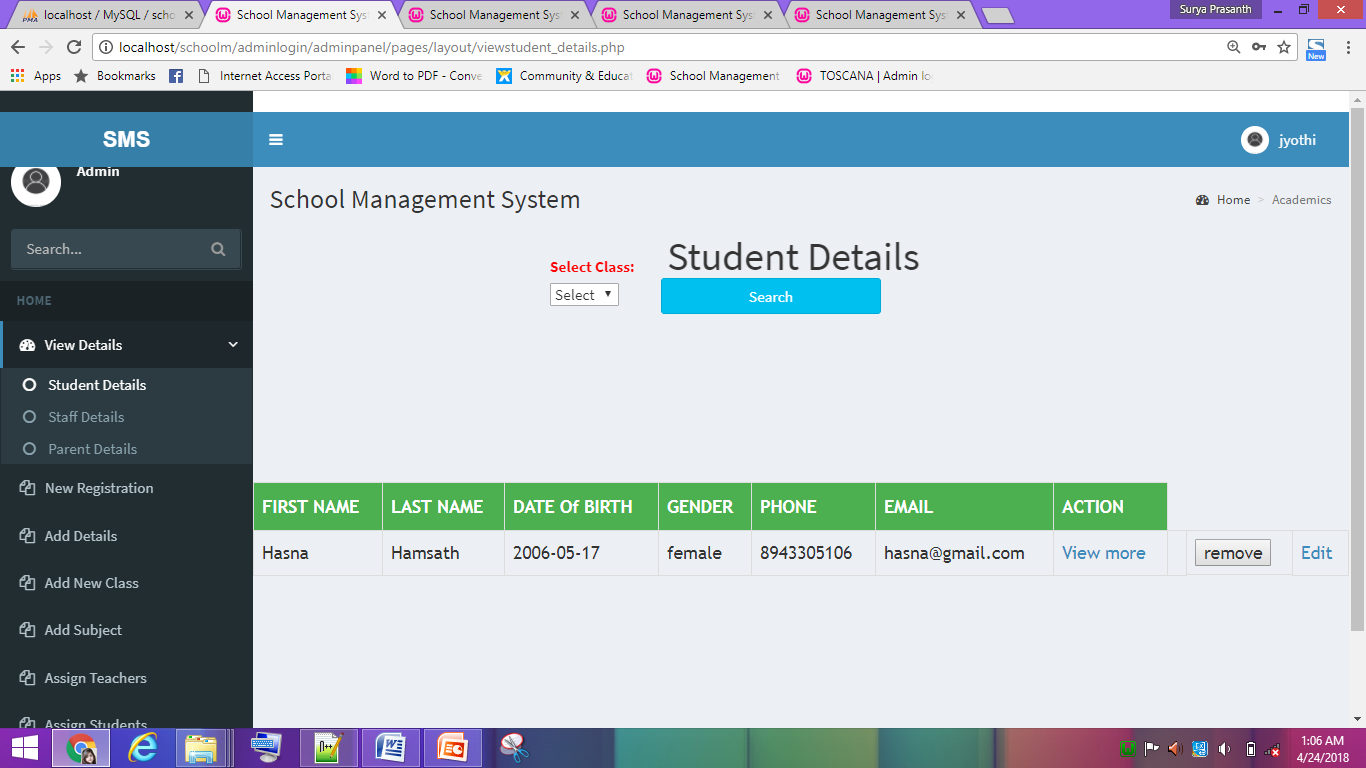
**Figure 18: Admin Dashboard**

****

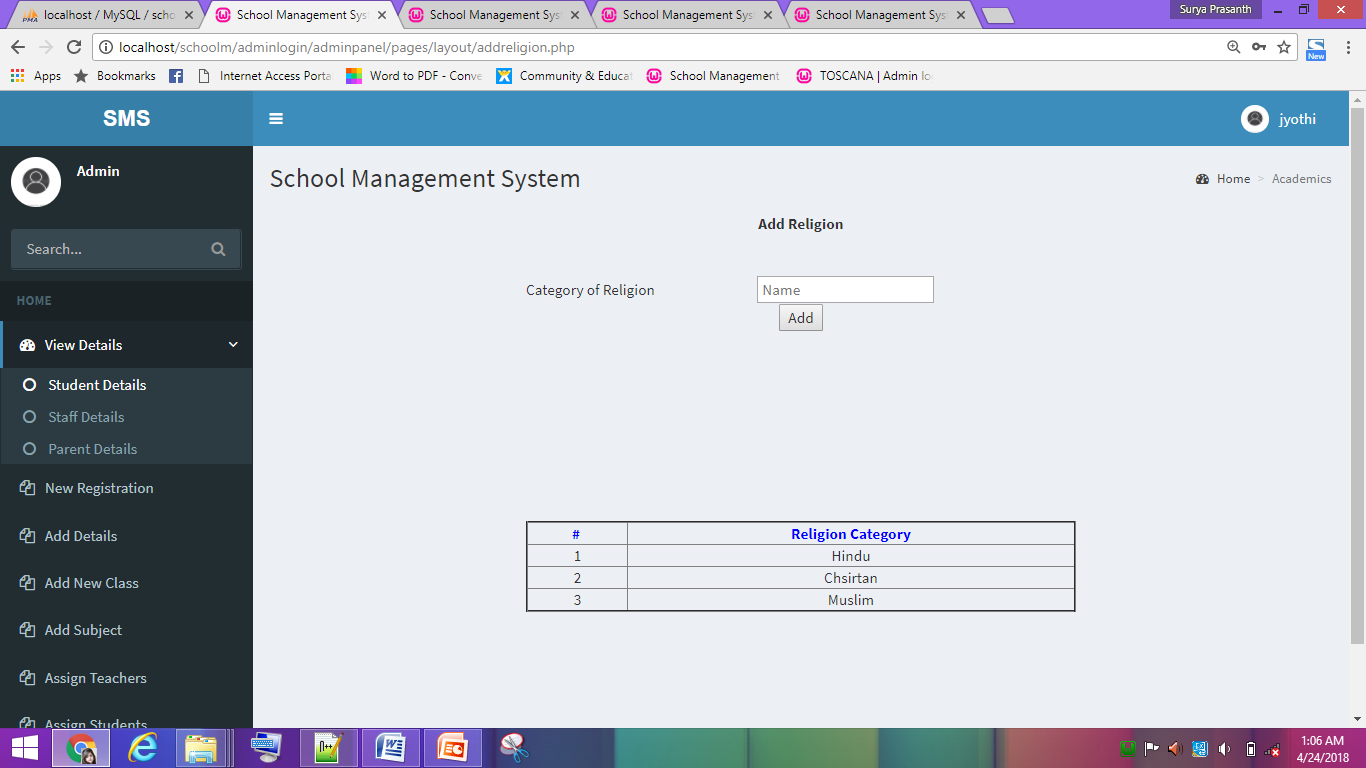
**Figure 19: Student Registration page**

****

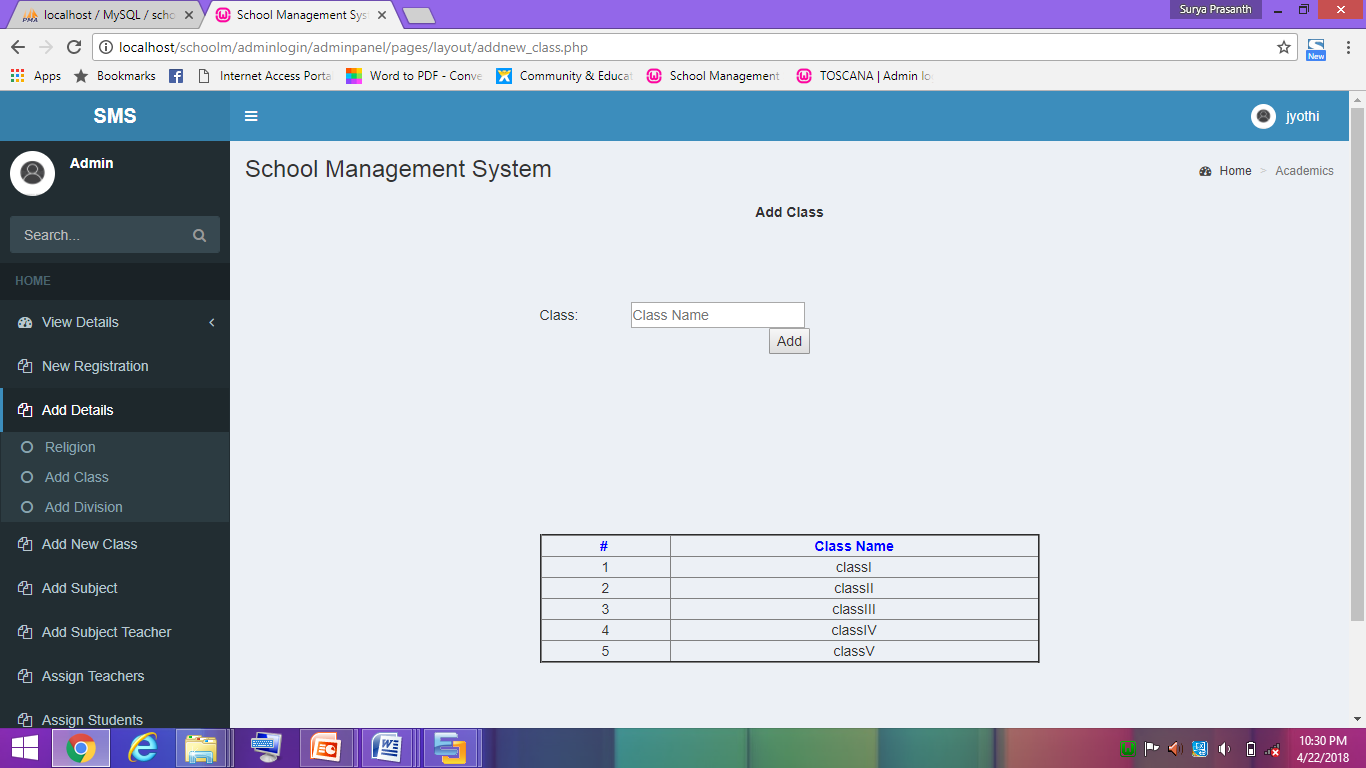
**Figure 20: Staff Registration page**

****

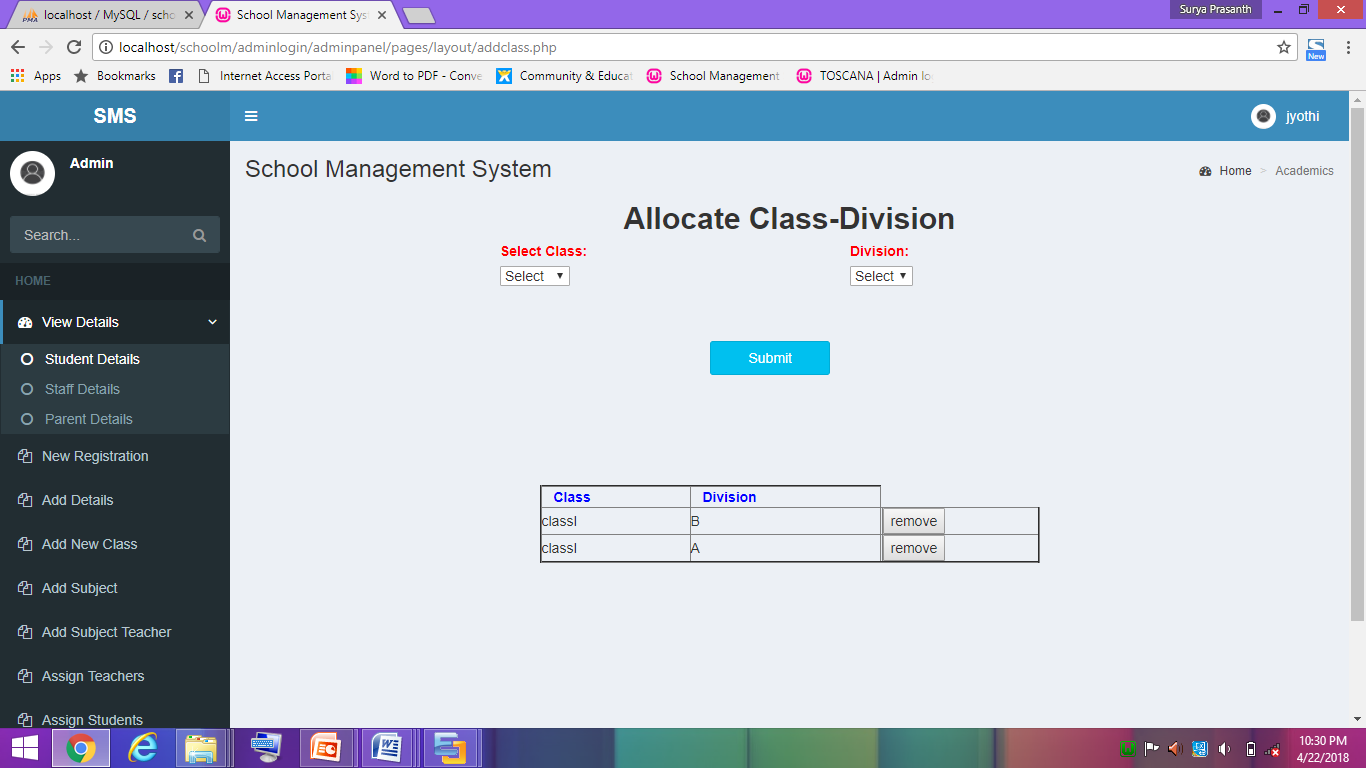
**Figure 21: view student details page**

****

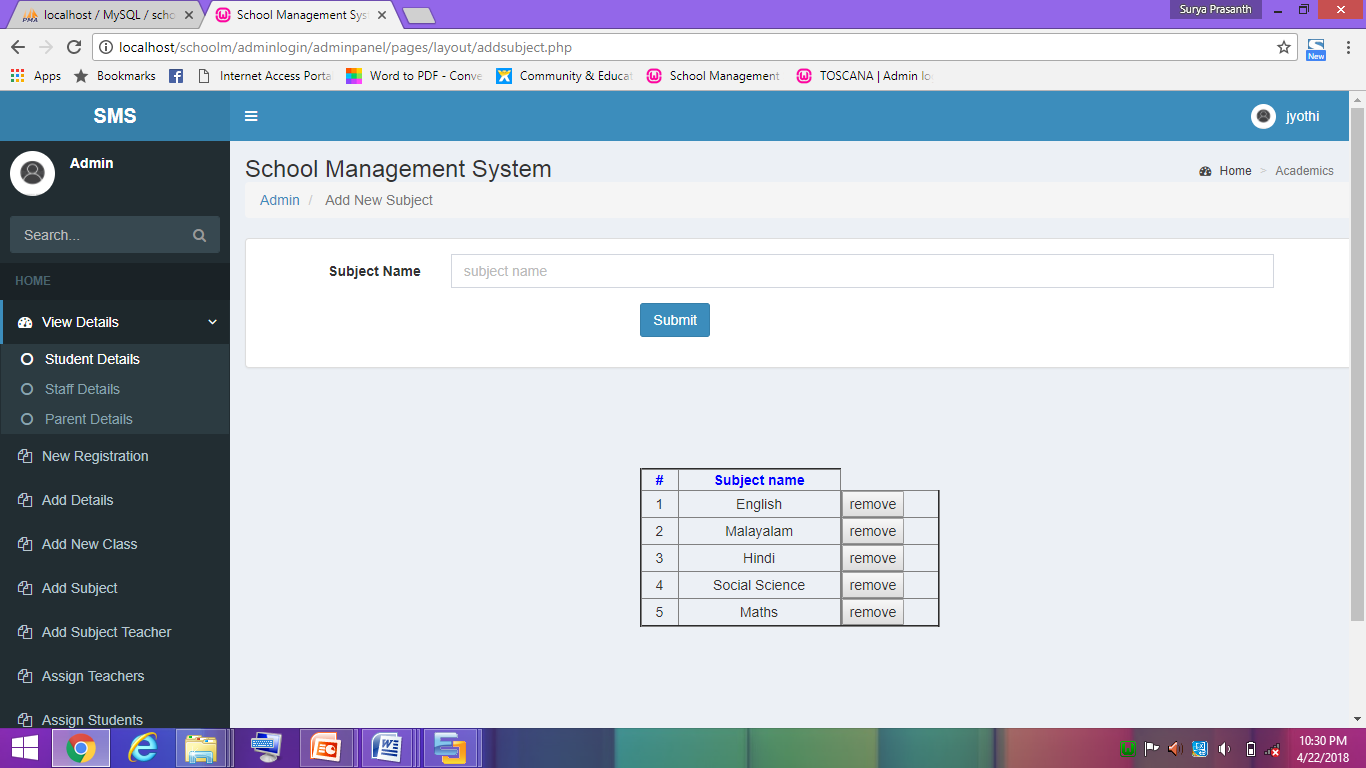
**Figure 22: add new religion**

****

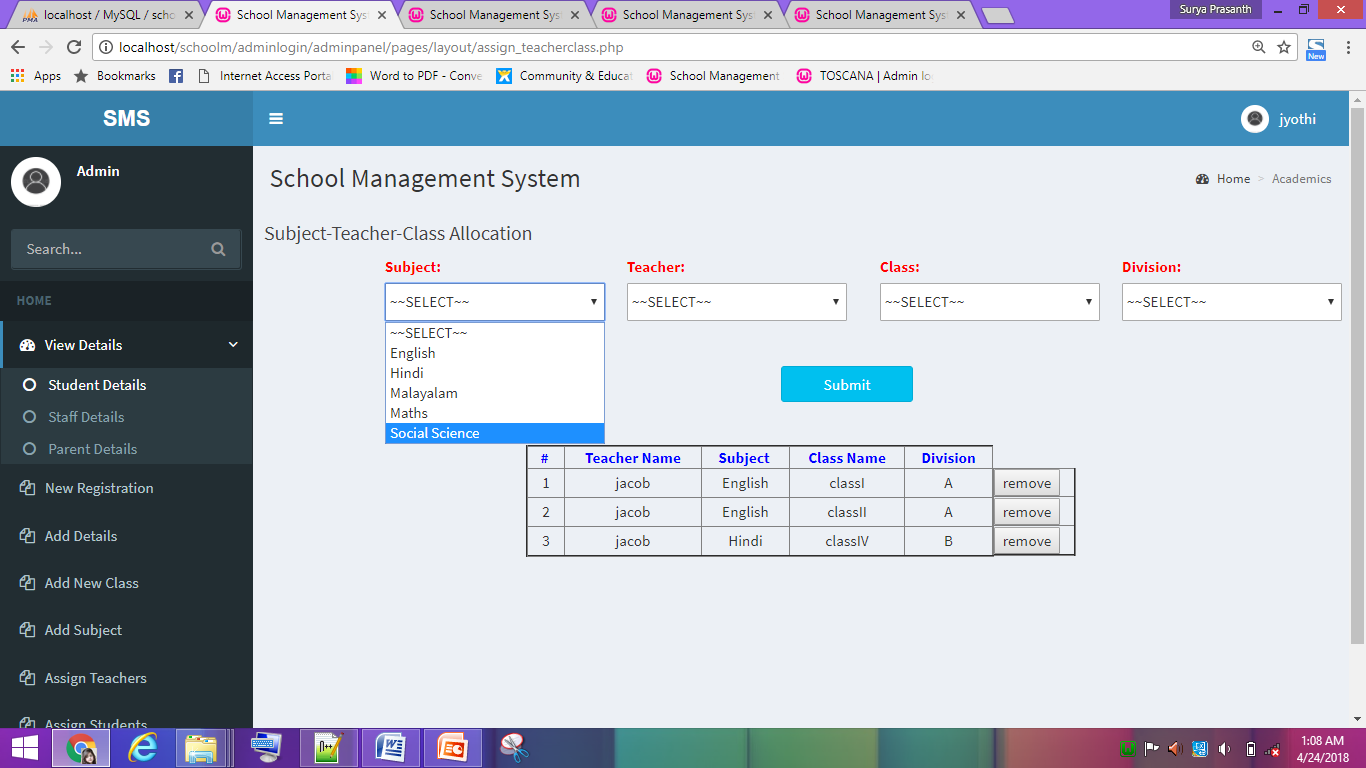
**Figure 23: add new class**

****

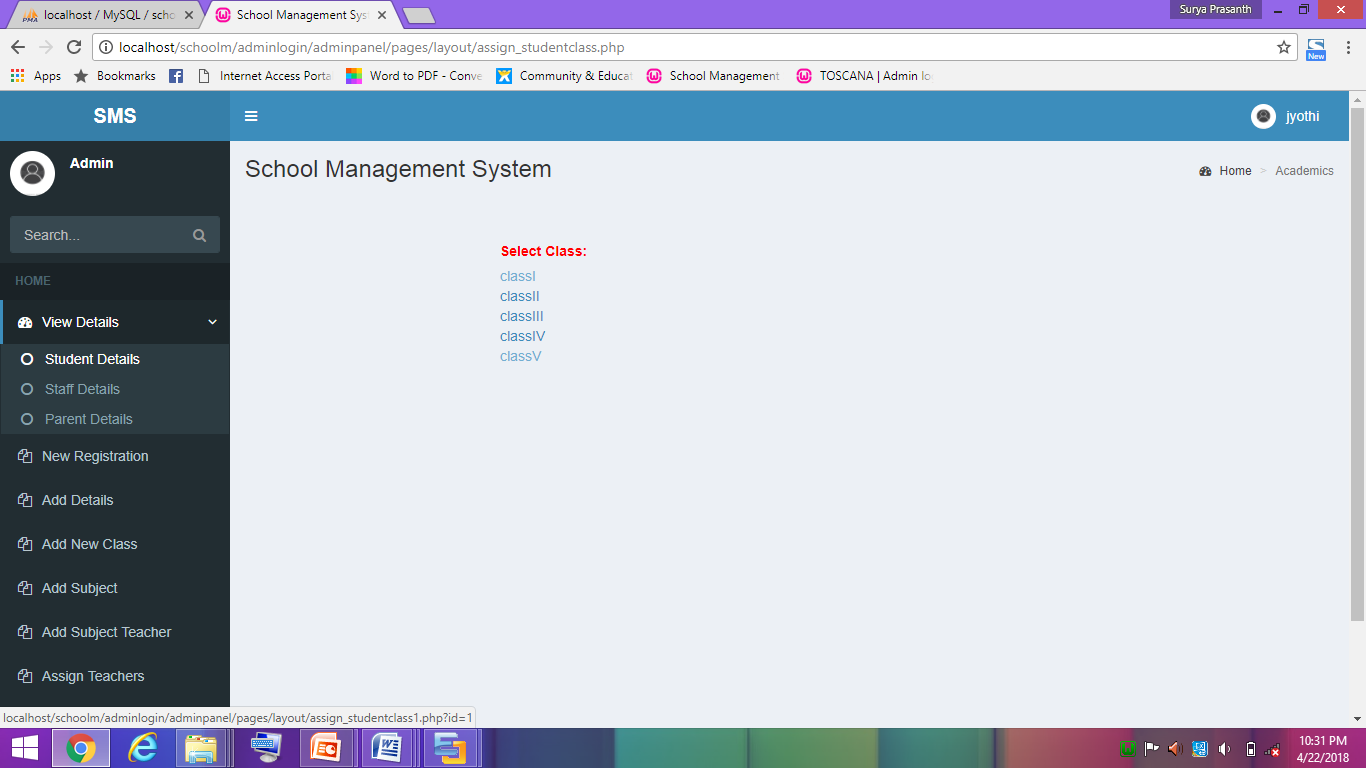
**Figure 24: allocate class and division**

****

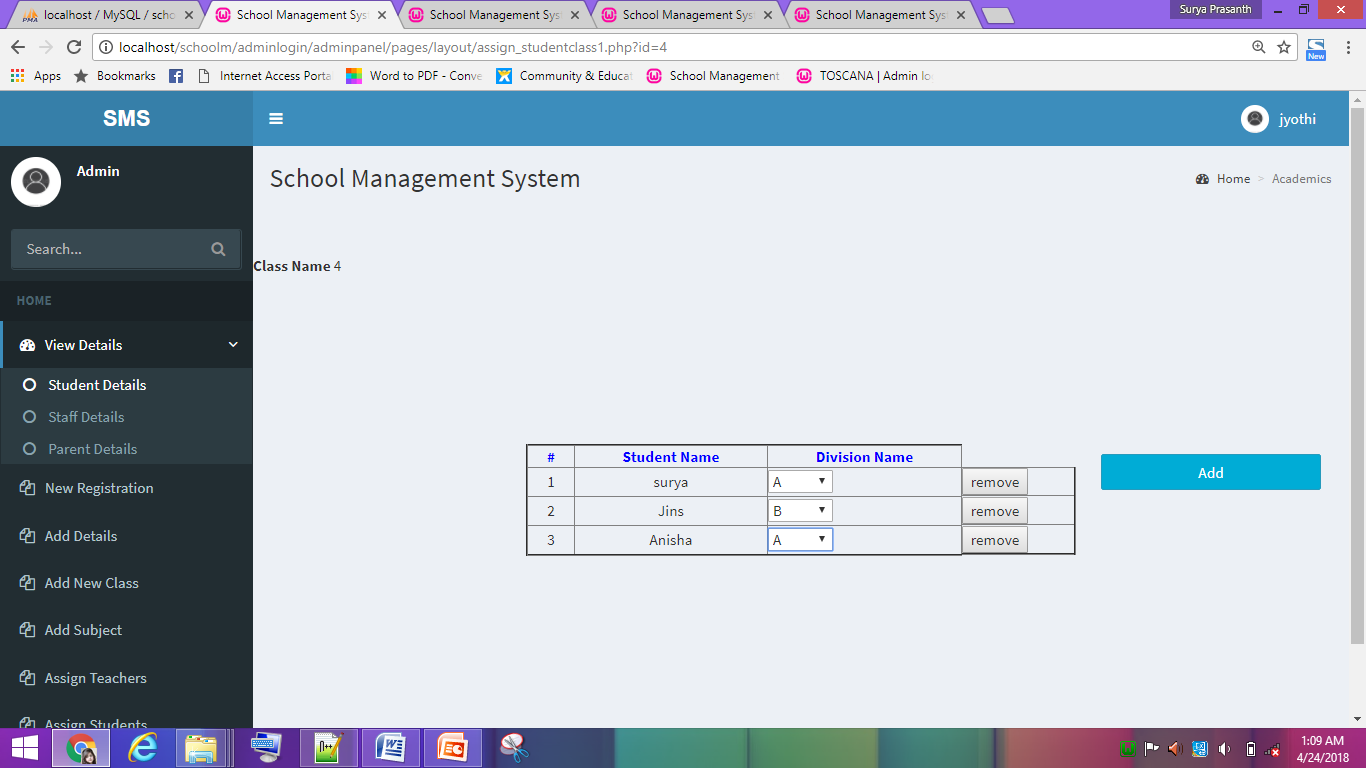
**Figure 25: add new subject**

****

**Figure 26: subject-teacher-class-division allocation**

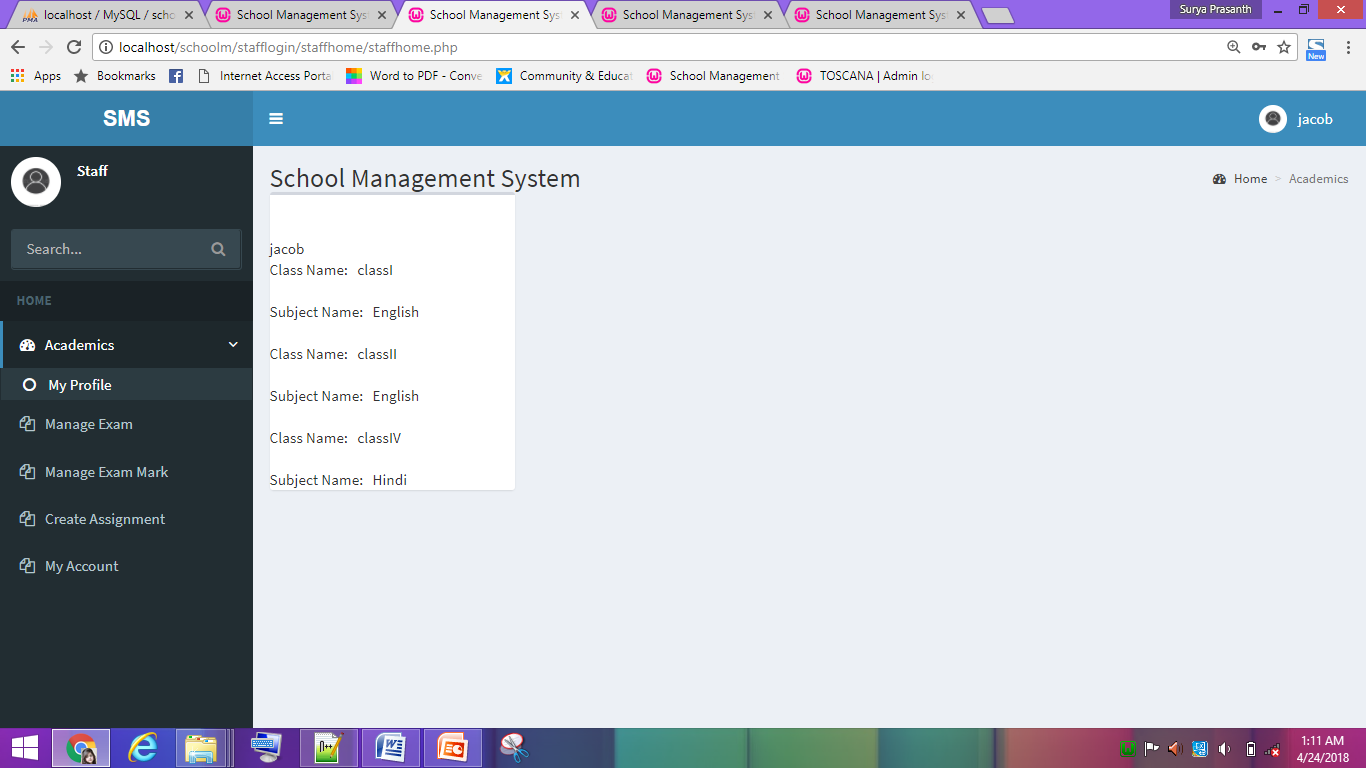
****

**Figure 27: student-class allocation**

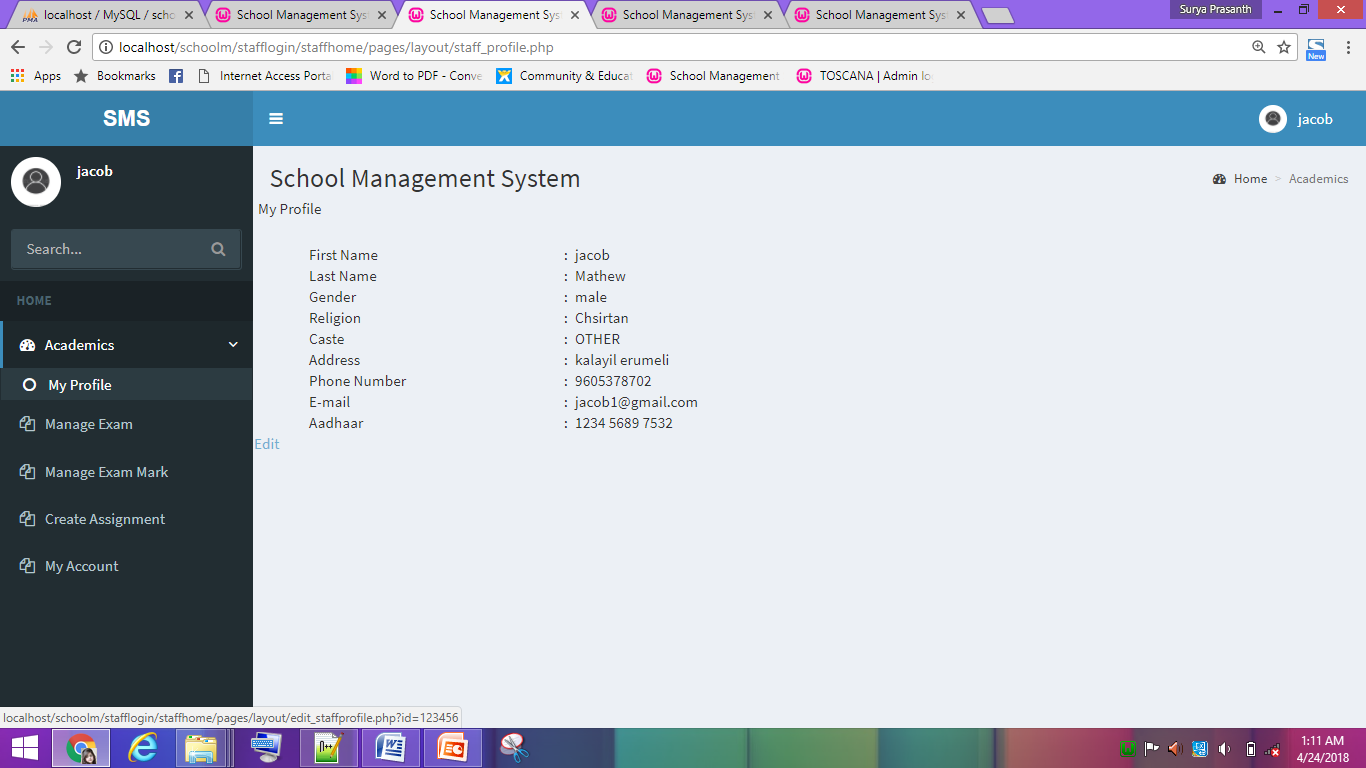
****

**Figure 28: student-class-division allocation**

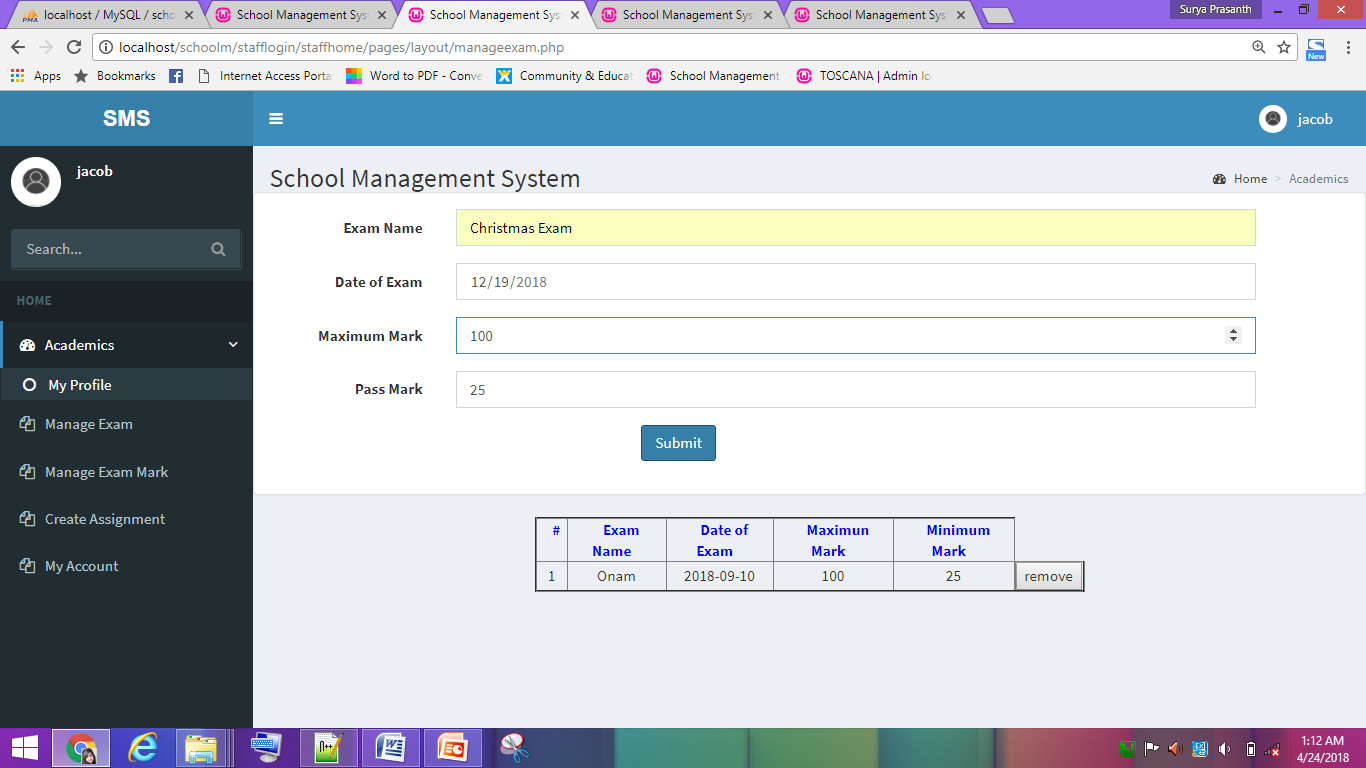
**TEACHER**

****

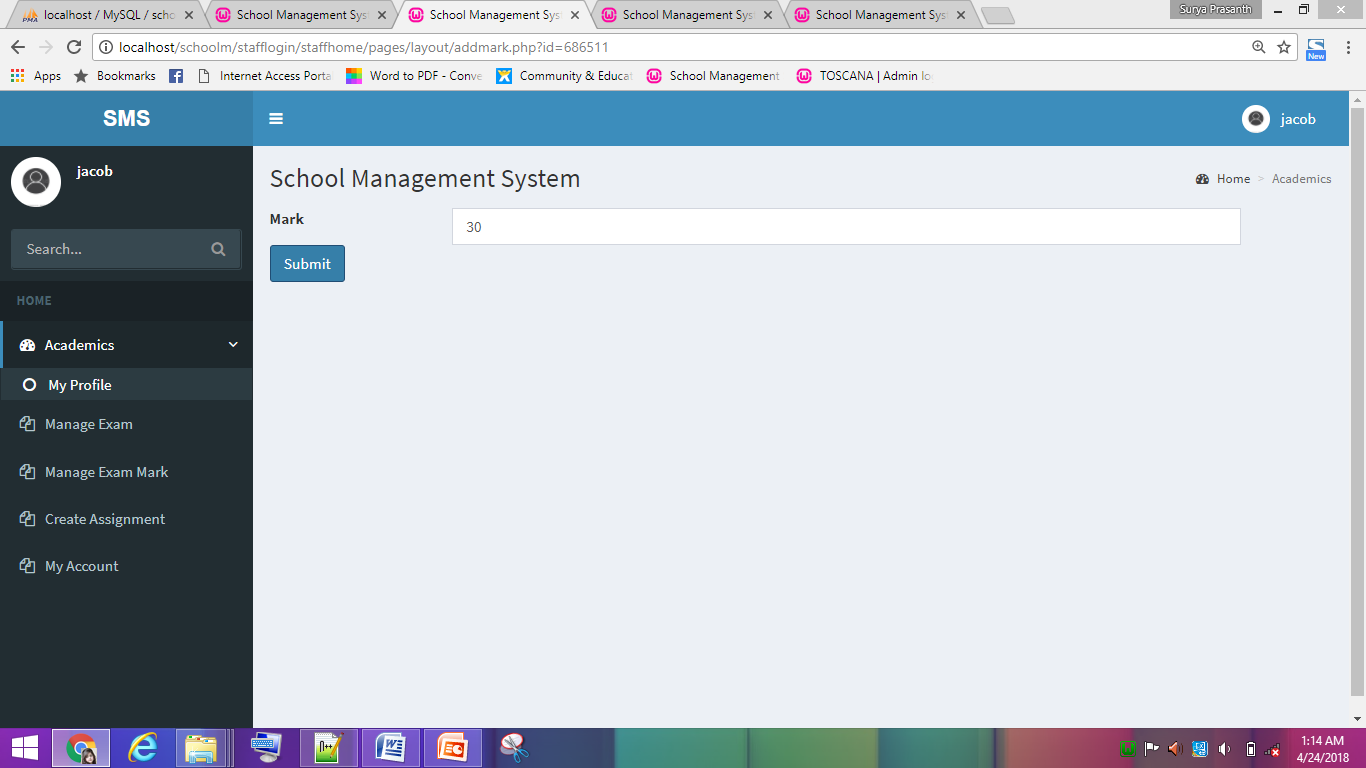
**Figure 29: teacher dashboard**

****

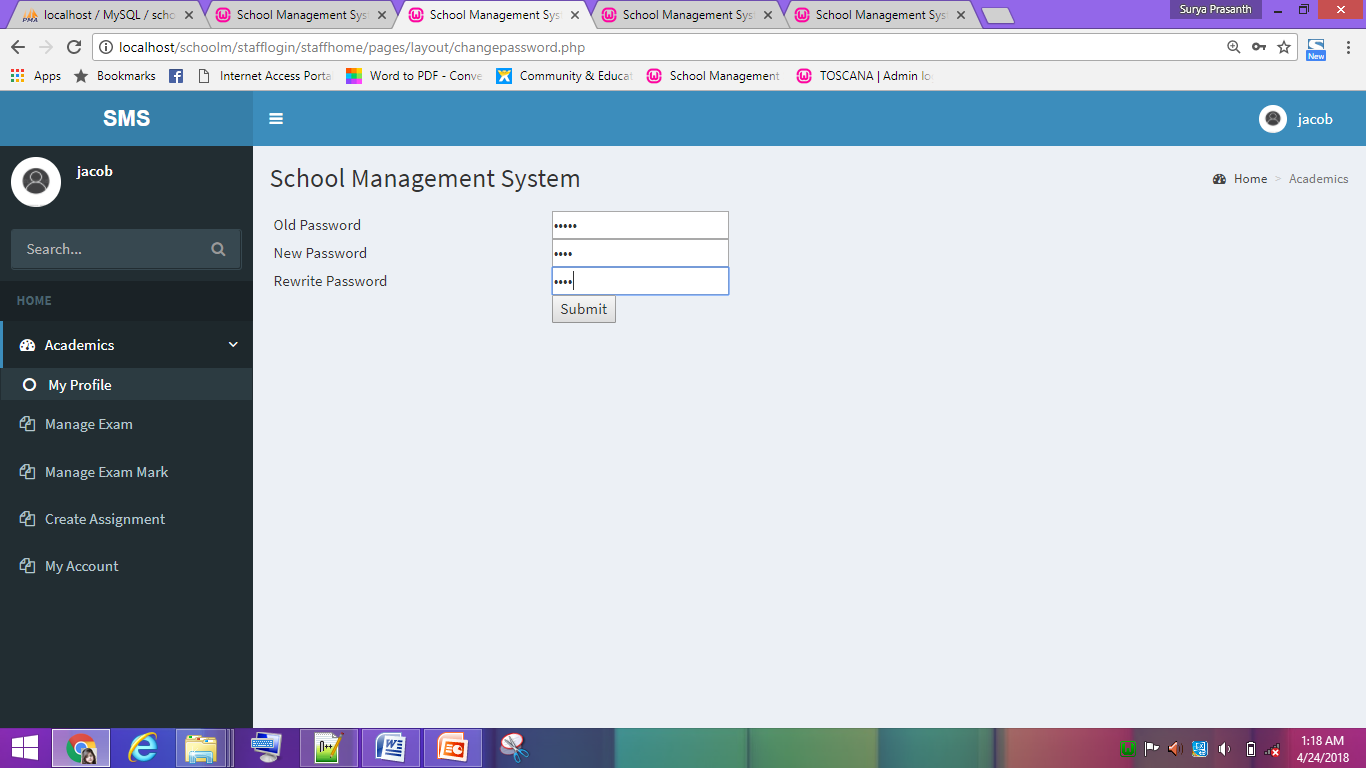
**Figure 30: teacher profile**

****

**Figure 30: manage exam**

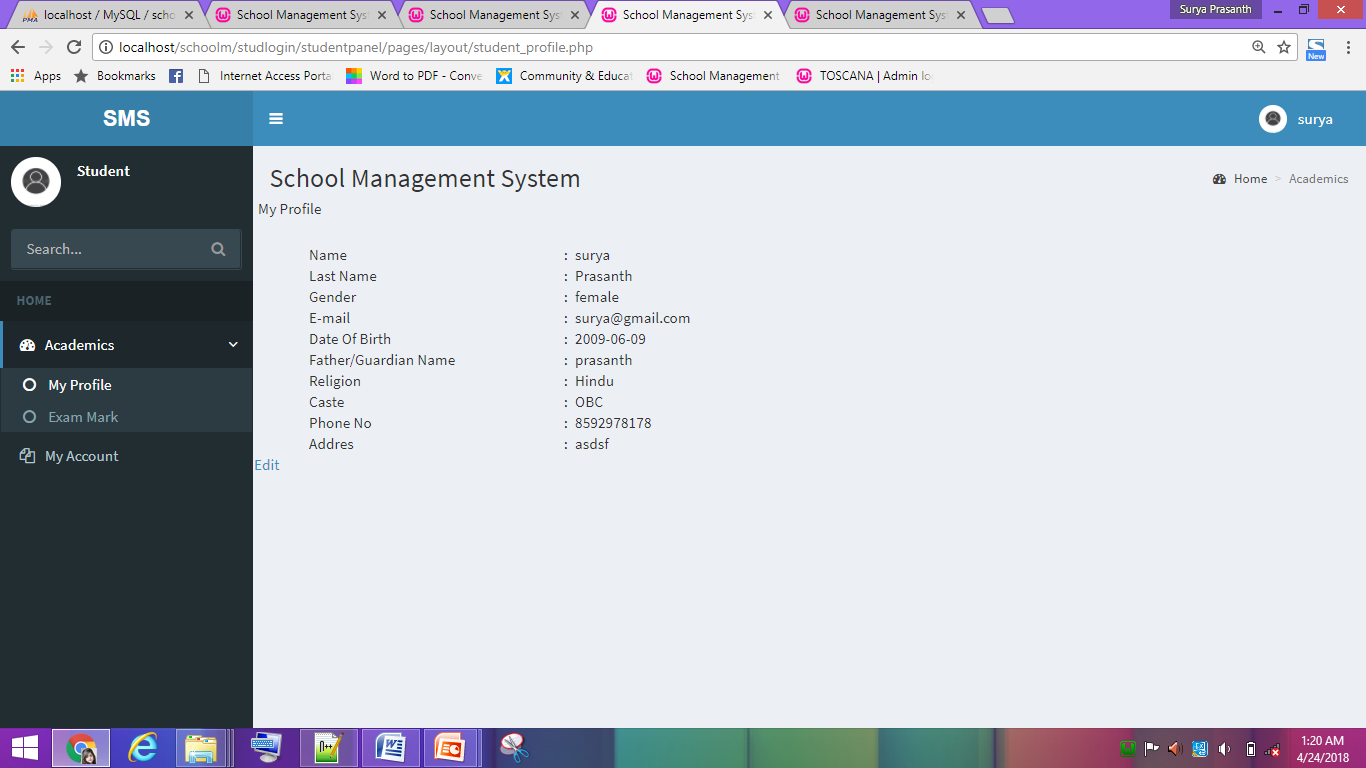
****

**Figure 31: manage mark**

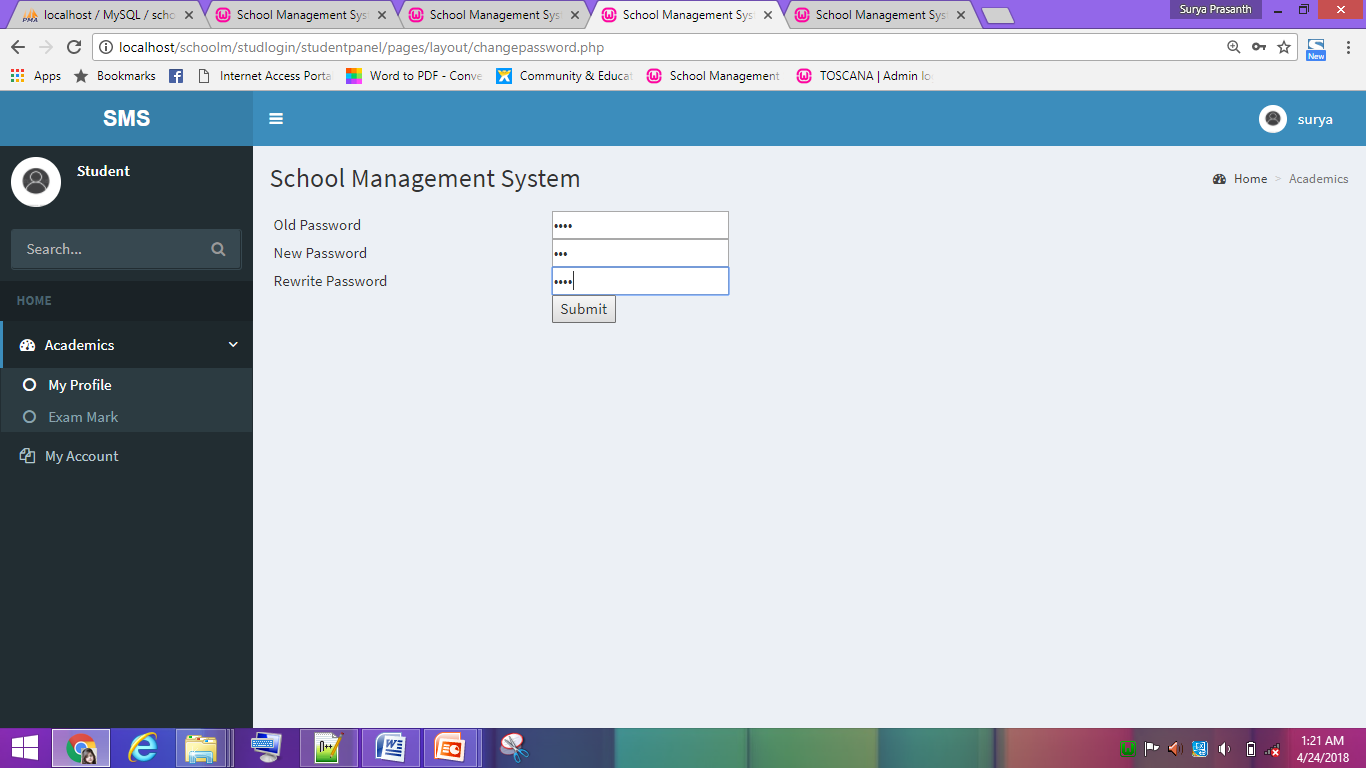
****

**Figure 32: change password**

**STUDENT**

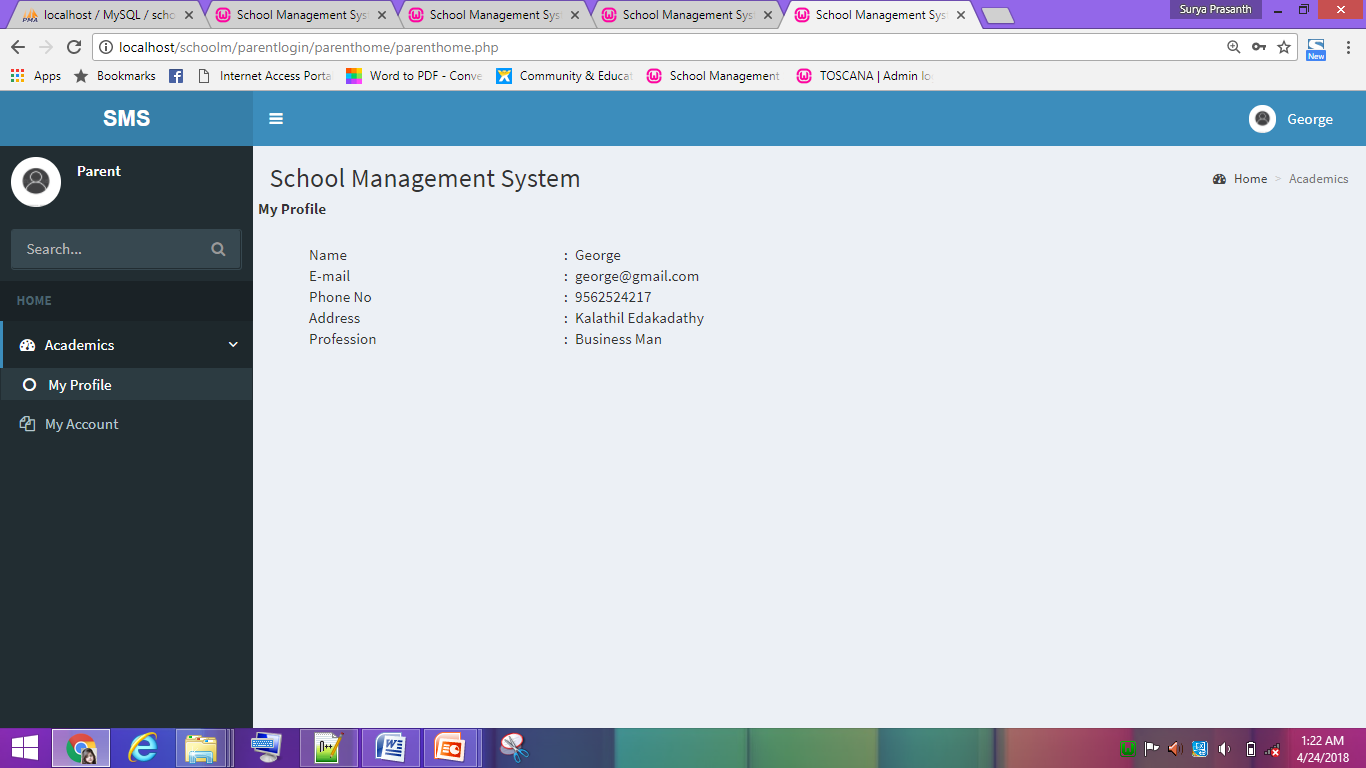
****

**Figure 33: student profile**

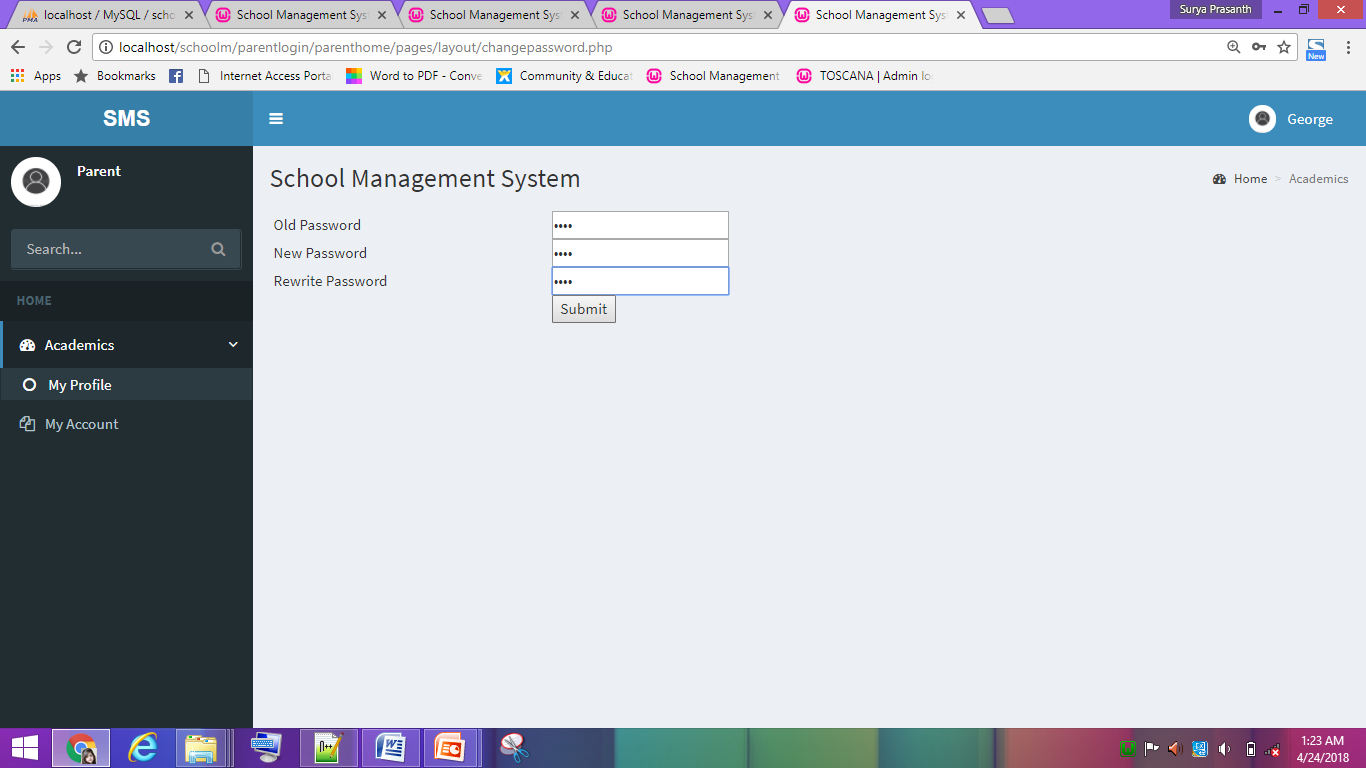
****

**Figure 34: change password**

**PARENT**

****

**Figure 35: parent profile**

****

**Figure 36: change password**

**4.7 DATA BASE DESIGN**

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS. In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

• Data Integrity

• Data independence.

**4.6.1 Relational Database Management System (RDBMS)**

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values.

**Relationships**

• Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.

• Entity Integrity enforces that no Primary Key can have null values.

• Referential Integrity enforces that no Primary Key can have null values.

• Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other keys are Super Key and Candidate Keys.

**4.6.2 Normalization**

Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. Normalization eliminates redundancy and promotes integrity.

**First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. In this, we form new relations for each non-atomic attribute or nested relation.

**Second Normal Form**

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

**Third Normal Form**

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute

**TABLE DESIGN**

**Table No 1 : adminlog**

**Primary Key: username**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **description** |
| username | varchar | Primary key |
| Password | varchar | password |
| Status | int | status |

**Table no 2: studreg**

Primary key : studid

Foreign Key : username

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **description** |
| studid | int | Primary key |
| username | varchar | Foreign key |
| Name | varchar | Name |
| Lname | varchar | lastname |
| dob | date | Date of birth |
| sex | varchar | Sex |
| religion | varchar | Religion |
| caste | varchar | Caste of the student |
| addr | varchar | Address |
| phone | varchar | Phone number |
| email | varchar | Email |
| fname | varchar | Father’s name |
| status | Int | Status |

**Table no 3: teachreg**

Primary Key: teachid

Foreign Key : username

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| teachid | int | Primary key |
| username | varchar | Foreign key |
| teach\_name | varchar | Name |
| lastname | varchar | Lastname |
| sex | varchar | Sex |
| religion1 | varchar | Religion |
| caste | varchar | Caste of the teacher |
| addrt | varchar | Address |
| mob | varchar | Mobile |
| email | varchar | Email |
| aadhar | varchar | Aadhar card number |
| status | int | Status |

**Table no 4: parentreg**

**Primary Key: pid**

**Foreign Key: username**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| pid | int | Primary key |
| username | varchar | Foreign key |
| name | varchar | Name |
| email | varchar | Email |
| phone | varchar | Phone number |
| addrp | varchar | Address |
| prof | varchar | Profession |
| status | int | Status |

**Table no 5 : class**

**Primary Key: classid**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| classid | int | Primary key |
| clsname | varchar | Name |
| Status | int | Status |

**Table no 6: division**

**Primary Key: divid**

**Foreign Key:classid,diid**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| Divid | int | Primary key |
| Classid | int | Foreign key |
| Diid | int | Foreign key |
| Status | int | Status |

**Table no 7:subject**

**Primary Key: subid**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| subid | int | Primary key |
| name | varchar | Name |
| status | int | Status |

**Table no 8:substaff**

**Primary Key:sid**

**Foreign Key :subid,teachid**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| sid | int | Primary key |
| subid | int | Foreign key |
| teachid | int | Foreign key |
| status | int | Status |

**Table no 9: studentclass**

**Primary Key:stud\_classid**

**Foreign Key :studid, classid**

|  |  |  |
| --- | --- | --- |
| **Field** | **datatype** | **Description** |
| Stud\_classid | int | Primary key |
| studid | int | Foreign key |
| classid | int | Foreign key |
| Status | int | Status |

Table no 10:teachsubclass

**Primary Key :tscid**

**Foreign Key:teachid,subid,classid,diid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| tscid | int | Primary key |
| teachid | int | Foreign key |
| subid | int | Foreign key |
| classid | int | Foreign key |
| diid | int | Foreign key |
| status | int | Status |

**Table no 11: mexam**

**Primary Key:examid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| examid | int | Primary key |
| name | varchar | Name of the exam |
| date | date | Date of exam |
| maxmark | varchar | Maximum mark |
| minmark | varchar | Minimum mark |
| satus | int | status |

**Table no 12:assignment**

**Primary Key:asigid**

**Foreign Key:subid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| asigid | int | Primary key |
| subid | int | Foreign key |
| classid | int | Foreign key |
| Title | varchar | Topic |
| datesub | date | Submission date |
| status | int | Status |

Table no 13:Cls

**Primary Key:clsid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| clsid | int | Primary key |
| clsname | varchar | classname |
| status | int | status |

**Table no 14:Divn**

**Primary Key:diid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| Diid | int | Primary key |
| dname | varchar | Division name |
| status | int | status |

**Table no 15:Religion**

**Primary Key:rid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **description** |
| Rid | int | Primary key |
| Rname | varchar | religion |
| Status | int | status |

**5. TESTING**

##### **5.1 INTRODUCTION**

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation:Are we doing the right job?

Verification:Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analysing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

* A good test case is one that has high possibility of finding an undiscovered error.
* A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

* For correctness
* For implementation efficiency
* For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

**5.2TEST PLAN**

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

**5.2.1 Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to

ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

**5.2.2 Integration Testing**

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

**5.2.3 Validation Testing**

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

**5.2.4 User Acceptance Testing**

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

* Input Screen Designs,
* Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**5.3 TEST CASE**

**5.3.1 Test Cases 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: SMS | | | | | | | |
| **SGS.Com Test Case 1** | | | | | | | |
| Test Case ID : Test Case 1 | | | | | Test Designed by: Surya Prasanth | | |
| Test Priority(Low/Medium/High):Medium | | | | | Test Designed date:20/1/2018 | | |
| Module Name: Login screen | | | | | Test Executed by: Ms Sona Sebastian. | | |
| Test Title: Verify login with valid username and Password | | | | | Test Execution date:20/1/2018 | | |
| Description: Test the SMS Login Page | | | | |  | | |
| Pre-conditions: User has Valid username and Password | | | | | | | |
| Dependencies | | | | | | | |
| Step | Test Steps | Test Data | Expected Result | Actual Result | | Status(Pass/Fail) | Note |
| 1 | Navigation to login page |  |  |  | | Pass |  |
| 2 | Provide Valid username. | Username admin | User  should be able to  login | Logged in and the User is navigated to User Dashboard with  Details | | Pass |  |
| 3 | Provide valid password | Password: admin |  |
| 4 | Click on login Button |  |  |  |
| 5 | Provide Invalid username and correct password | Username: admin2  Password: admin | User  Should not able to  login | Error message “Invalid login credentials” | | Pass |  |
| **Post conditions:**  User is validated with database and successfully login to account. The account session details are logged in database | | | | | | | |

**5.3.2 Test Cases 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: SMS.Com | | | | | | | |
| **SMS Test Case 2** | | | | | | | |
| Test Case ID:Test Case 2 | | | | | Test Designed by: Surya Prasanth | | |
| Test Priority(Low/Medium/High):Medium | | | | | Test Designed date:20/01/2087 | | |
| Module Name: Adding Staff profile details | | | | | Test Executed by: Sona Sebastian | | |
| Test Title: Adding the details user | | | | | Test Execution date:20/01/2018 | | |
| Description: SMS adding details of user | | | | |  | | |
| Pre-conditions: User data should be correct | | | | | | | |
| Dependencies | | | | | | | |
| Step | Test Steps | Test Data | Expected Result | Actual Result | | Status(Pass/Fail) | Note |
| 1 | Navigation to add staff page |  |  |  | | Fail |  |
| 2 | Provide  Valid registration details | Name, mobile, house name,  Username, password, | Successfully staff registered | Successful Message shown  and redirected to  admin dashboard. | | Pass |  |
| 3 | Click on Save Button | Validations |  |
| 4 | Required fields are not filled | Valid address, phone is provided, name and mail\_id is not providing | User should not be able to register | Mouse Focuses on the blank field | | Pass |  |
| **Post conditions:**  User is validated with database and other validations. | | | | | | | |

**5.4 TEST REPORT**

In SMS each module is tested and errors are rectified. The testing process is done by providing both valid and invalid data and examined how the system responds under these circumstances.

**Registration and Login**

In this module, all the fields are validated. The user should enter valid details. If an invalid data is given, then a message is displayed to notify them. In this module, for admin register staff and students and parents. The information’s such as Name, address, contact details, etc are entered into the database. After staff registration, they can log in to the school site using username and password. They can also update some of the personal information and password. When registration is completed, they will redirect to their Home page. During the registration process, all the inputs are validated and if there is any error, it will display on the screen itself. If a username is already in use, it will also display in the screen, so that he can choose another username. Students have their admission number as their username. To get in to the system, the users should need a valid username and password.

# 6.IMPLEMENTATION

**6.1 IMPLEMENTATION PROCEDURES**

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

* Careful planning.
* Investigation of system and constraints.
* Design of methods to achieve the changeover.
* Training of the staff in the changeover phase.

**6.1 IMPLEMENTATION PROCEDURES**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. Inmany organizations someone who will not be operating it, will commission the softwaredevelopment project. In the initial stage people doubt about the software but we have toensure that the resistance does not build up, as one has to make sure that:

* The active user must be aware of the benefits of using the new system.
* Their confidence in the software is built up.
* Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

**6.1.1 User Training**

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

**6.1.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across differentuser groups and across different levels of hierarchy.

**6.1.3 System Maintenance**

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

# 

# 

# 7.CONCLUION & FUTURE ENHANCEMENTS

**7.1 FUTURE ENHANCEMENT**

As a future enhancement planning this system can be expanded to the high school and upper primary sections. Also planning to develop an android application for the parent to see their child academic marks. Also planning for a transport management module which enables effective management of School Transport. You can assign Routes to specific transportation assets and map students to these routes. The mobile app helps in tracking the position of the bus using inbuilt GPS tracking systems. Also planning to integrate a fully functional payroll management system into the system to ease the burden on HR department. The payroll module is fully compliant with the rules as laid down by Law. Various calculation parameters and the allowances are also configurable as per the user requirements. As a next step planning for a platform for Alumni Interaction. Alumni can get connected with school, friends and teachers through the portal.

.

**7.2 CONCLUSION**

The software reduces the time consumption and the manual efforts of searching a service provider. It will be a simple platform for users to access services for their huge needs. The benefits, we can obtain from the new system are:

* Timely and accurate information will be available
* Reduced data loss
* The access time and process time is highly reduced
* Quick data view
* Error free output

The proposed system is expected to replace manual system and provide more efficient performance and services.It has been a great pleasure for me to work on this exciting and challenging project. The SMS will help students, teachers, parents and the school administrative staff to automate all the administrative tasks and use school data in a more organized and structured manner and enable paperless administrative implementation. It allows users to interact with basic operation and information of their schools effectively and also provides access to all relevant reports that have been generated over the years. It has been designed to provide an enhanced interaction between students, teachers, parents and the management and can be accessed over the internet from any location in the world. The system is developed in a manner that it is user friendly and required help is provided at different levels of users.

**8.BIBILOGRAPHY**

**BOOKS/REFERENCES:**

* Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
* Roger S Pressman, “*Software Engineering*”, 1994.
* PankajJalote, “So*ftware engineering*: a precise approach”, 2006.
* James lee and Brent ware Addison, “Open source web development with LAMP”,2003
* IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

* www.w3schools.com
* www.jquery.com
* http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
* www.agilemodeling.com/artifacts/useCaseDiagram.htm
* www.stackoverflow.com
* www.slideshare.net

# 9.APPENDIX

##### **9.1 SAMPLE CODE**

<?php

include\_once "C:\wamp64\www\schoolm\adminlogin\adminpanel\pages\layout\header.php";?>

<html><head><meta charset="UTF-8">

<title>school Management System</title>

<head>

<script src="js/reg\_validate.js"></script>

<link rel="stylesheet" href="css/style.css" type="text/css"><body>

<div id="backg" style="float:left;background-color:solid white;font-size:23;font-color:black;width: 1000px;height: 100px;">

<h1><marquee direction="left">New Student Register here</marquee></h1>

<form name="myform" method="post" action="">

<table border="0" width="500" align="center" class="demo-table">

<!--tr><center><b>Register Here</b></center><br></tr-->

<script src="js/reg\_validate.js"></script>

<!--tr><td>Student id</td>

<td><input type="text" name="studid" placeholder="studid" pattern="[0-5]{4}" title="Must Contain numbers" required ></td>

</tr--><tr>

<td> <b>First Name</b></td>

<td><input placeholder="First Name" type="text" id="fname" name="name" onchange="f()" required="" ></td>

</tr><tr>

<td> <b>Last Name</b></td>

<td><input placeholder=" Last Name" type="text" id="lname" name="lnam" onchange="f()" required="" ></td>

</tr><tr>

<td><b>Date of Birth</b></td>

<td><input type="date" id="dob" name="dob" min="2000-04-01" max="2012-04-20" required>

</tr><tr>

<td><b>Gender</b></td>

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

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

</td></tr>

<tr align="left" >

<td class="narmal"><b>Class</b></td> <td>

<select name="classn" class="span8 tip" required >

<option value="">Select </option>

<?php

$con1=mysqli\_connect("localhost","root","","school");

$query=mysqli\_query($con1,"select \* from class");

while($row=mysqli\_fetch\_assoc($query))

{?><option value="<?php echo $row['classid'];?>"><?php echo $row['clsname'];?></option>

<?php } ?>

</select></td></tr>

<tr align="left" >

<td class="narmal"><b>Religion</b></td> <td>

<select name="religi" class="span8 tip" required >

<option value="">Select </option>

<?php

$con1=mysqli\_connect("localhost","root","","school");

$query=mysqli\_query($con1,"select \* from religion");

while($row=mysqli\_fetch\_assoc($query))

{?><option value="<?php echo $row['rid'];?>"><?php echo $row['rname'];?></option>

<?php } ?></select></td></tr>

<tr ><td class="narmal"><b>Caste</b></td>

<td><select name="caste1" id="caste1" required=required>

<option value="">Select</option>

<option value="SC">SC</option>

<option value="ST">ST</option>

<option value="OBC">OBC</option>

<option value="VJ/NT">VJ/NT</option>

<option value="SBC">SBC</option>

<option value="OTHER">OTHER</option>

</select></td> </tr><tr>&nbsp;

<td><b>Address</b></td>

<td><textarea name="address" placeholder="Address" pattern="[A-Za-z]{1,32}" title="Must Contain Characters" required></textarea></td>

</tr><tr>

<td><b>Phone Number</b></td>

<td><input type="text" name="phone" id="phone" placeholder="Phone Number" onchange="p()" required></td>

</tr><tr><td><b>Email</b></td>

<td><input type="text" id="mail" name="email" placeholder="email" onchange="ml()" required></td></tr><tr><td><b>Name of Father</b></td>

<td><input type="text" name="fname" placeholder="name of father" title="Must Contain Characters" onchange="f()" required ></td></tr><tr>

<td><b>User Name</b></td>

<td><input type="text" name="username" placeholder="username" pattern="[0-9]{6}" title="Must Contain numbers" required ></td></tr><tr>

<td><b>Password</b></td>

<td><input type="password" id="pass" class="demoInputBox" name="password" placeholder="Password" onchange="a()" required ></td></tr><tr><tr>

<tdcolspan=15><inputtype="submit"name="submit1"value="Register" class="btnRegister"></td>

</tr></table></form></div></body></html>

<?php

$con=mysqli\_connect("localhost","root","","school");

if(isset($\_POST['submit1']))

{

$name =$\_POST["name"];

$lname1 =$\_POST["lnam"];

$dob=$\_POST["dob"];

$sex=$\_POST["gender"];

$religion=$\_POST["religi"];

$caste=$\_POST["caste1"];

$classna=$\_POST["classn"];

$add=$\_POST["address"];

$ph=$\_POST["phone"];

$email=$\_POST["email"];

$fname=$\_POST["fname"];

$uname=$\_POST["username"];

$pass=$\_POST["password"];

$t=2;

$s=1;

if(!$con)

{

echo "Could not connect..Try again";

}else

{

$r1="insert into adminlog(username,password,typeid,status) VALUES('$uname','$pass',$t,$s)";

mysqli\_query($con,$r1);

$rr="insert into studreg(username,name,lname,dob,sex,religion,caste,addr,phone,email,fname,status)values('$uname','$name','$lname1','$dob','$sex','$religion','$caste','$add','$ph','$email','$fname',$s)";

mysqli\_query($con,$rr);

$r2="select studid from studreg where username='$uname'";

$r1=mysqli\_query($con,$r2);

$rw=mysqli\_fetch\_array($r1);

$r3="insert into studentclass(`studid`,`classid`,`status`) values($rw[0],$classna,$s)";

mysqli\_query($con,$r3);

if($rr)

{

?><script>alert("registration success");</script>

<?php

}}

mysqli\_close($con);

}?>