Chapter 1

INTRODUCTION

1.1 OVERVIEW

Student Information Management System (SIMS) is a fully computerized system or a database where all the student related data can be stored, retrieved, monitored & analysed. The data is saved at a centralized location & role-based login access is given to all the stakeholders for ensuring student data security.

Our 'Student Information Management System' aids in managing, storing, data such as admission number, tracking attendance, exam & test marks, notice board, teacher info, subject info and other student related data. The system gives both the student and teacher an overall view of the ongoing academic activities.

1.2 PROBLEM STATEMENT

Students form the main part of any institution that concerns with. But the institution finds it difficult to keep the details of all the students. Thus, it involves a lot of paper work. Sometimes there will be huge heap of files bundled up and must be stored in some corner of the office. Managing student records manually is a troublesome job, this increases as the number of students increases. This method may lead to missing or destroying important information and is also difficult to maintain. Since this method of managing student data is outdated, we propose a better way, i.e., using a computer software which can automate most of the work in maintaining such records.

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end user’s application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things:

- Data
- Database engine, that allows data to be accessed, locked and modified.
- Database schema, which defines the database’s logical structure.

These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning, backup and recovery. 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.

1.4 STRUCTURED QUERY LANGUAGE (SQL)

SQL is a standard language for storing, manipulating and retrieving data in databases. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

Following are some interesting facts about SQL.

- SQL is case insensitive. But it is a recommended practice to use keywords (like SELECT, UPDATE, CREATE, etc) in capital letters and use user defined things (liked table name, column name, etc) in small letters.
- We can write comments in SQL using "–" (double hyphen) at the beginning of any line.
- SQL is the programming language for relational databases (explained below) like MySQL, Oracle, Sybase, SQL Server, etc. Other non-relational databases (also called NoSQL) databases like MongoDB, DynamoDB, etc do not use SQL
- Although there is an ISO standard for SQL, most of the implementations slightly vary in syntax. So, we may encounter queries that work in SQL Server but do not work in MySQL.

1.5 HYPERTEXT MARKUP LANGUAGE (HTML)

HTML is a standard mark-up language used for structuring and presenting content on the World Wide Web It is the fifth and last major HTML version that is a World Wide Web Consortium (W3C) recommendation. HTML5 includes detailed processing models to encourage more interoperable implementations, it extends, improves and rationalizes the mark-up available for documents, and introduces mark-up and application programming interfaces (APIs) for complex web applications.

1.6 JAVASCRIPT

JavaScript (JS) is a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. Alongside HTML and CSS, JavaScript is one of the third core technologies of the World Wide Web.
JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user's browser, etc.

JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

1.7 PHP

PHP is a recursive acronym for "PHP: Hypertext Pre-processor". PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

1.8. XAMPP

XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standard.

Chapter 2

REQUIREMENTS SPECIFICATIONS

A computerized way of handling information about property and users' details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

2.2.1 SOFTWARE REQUIREMENTS

• Database Support	: MySQL 5.7
• Operating system	: ANY OS (Recommended**:** Windows XP or above)
• Web Browser	: Google Chrome 60 or later
• Coding language	: PHP, HTML, CSS, JS.
• Server Deployment	: Apache Server, MySQL Server.

2.2.2 HARDWARE REQUIREMENTS

• CPU	: Pentium IV 2.4 GHz or above.
• Memory (Primary)	: 1 GB or above.
• Hard Disk	: 40 GB, 80GB, 160GB or above.
• Monitor	: 15 VGA color.
• Keyboard and Mouse	

2.2.3 TECHNOLOGY

- HTML is used for the front-end design. It provides a means to structure text-based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML, the language can be applied to any XML document.
- SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking. PHP Syntax is C-Like.
- JavaScript (JS) is a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code.

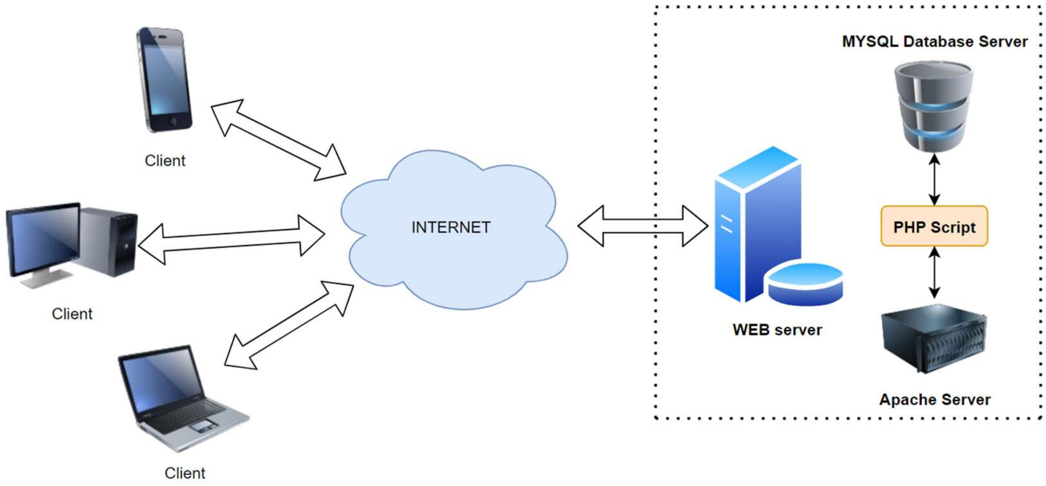
Chapter 3

DETAILED DESIGN

3.1 SYSTEM DESIGN

Apache HTTP Server is a free and open-source web server that delivers web content through the internet. Apache is just one component that is needed in a web application stack to deliver web content. One of the most common web application stacks involves LAMP, or Linux, Apache, MySQL, and PHP.

Linux is the operating system that handles the operations of the application. Apache is the web server that processes requests and serves web assets and content via HTTP. MySQL is the database that stores all your information in an easily queried format. PHP is the programming language that works with apache to help create dynamic web content.



Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language.

3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

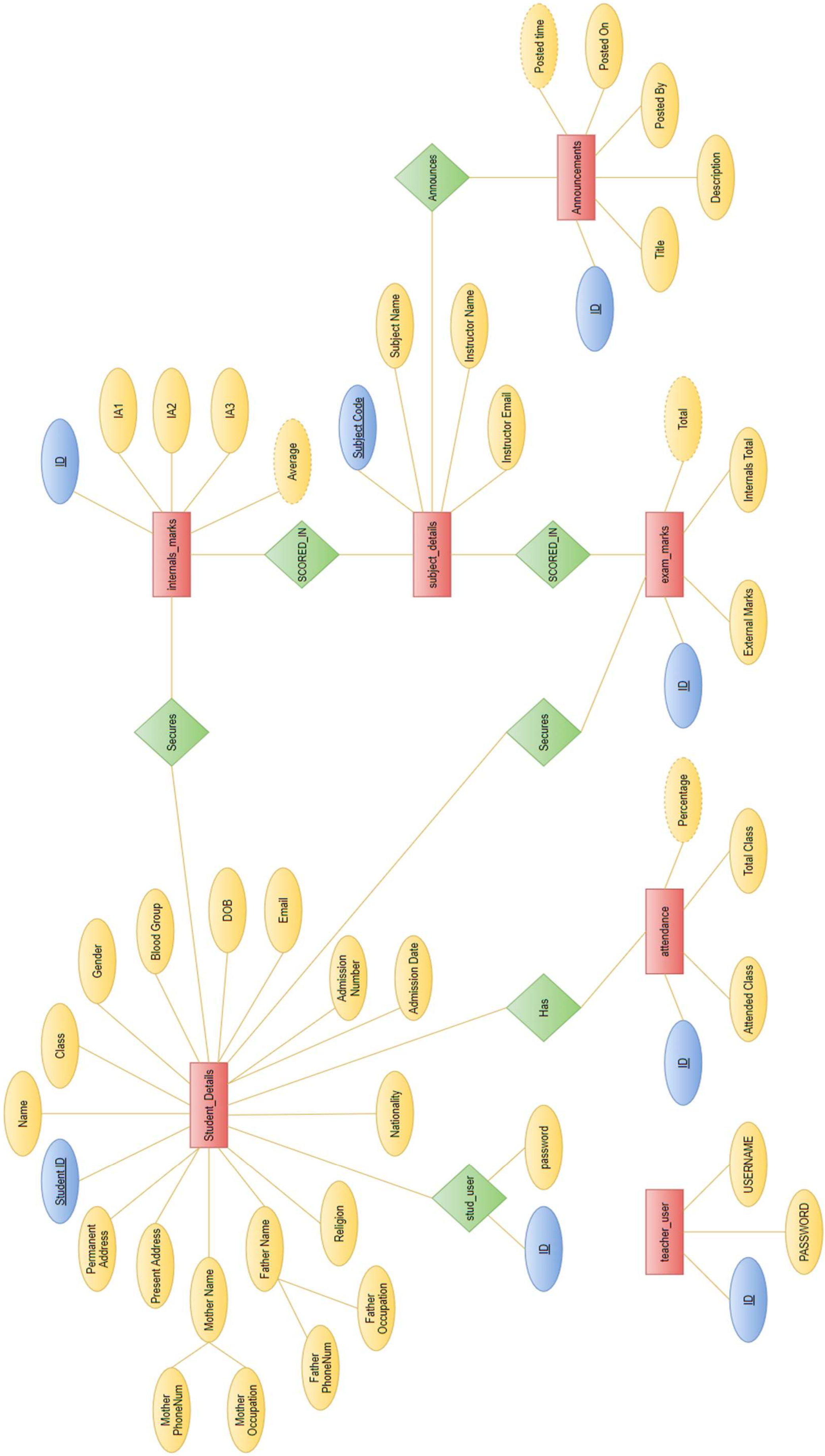
An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

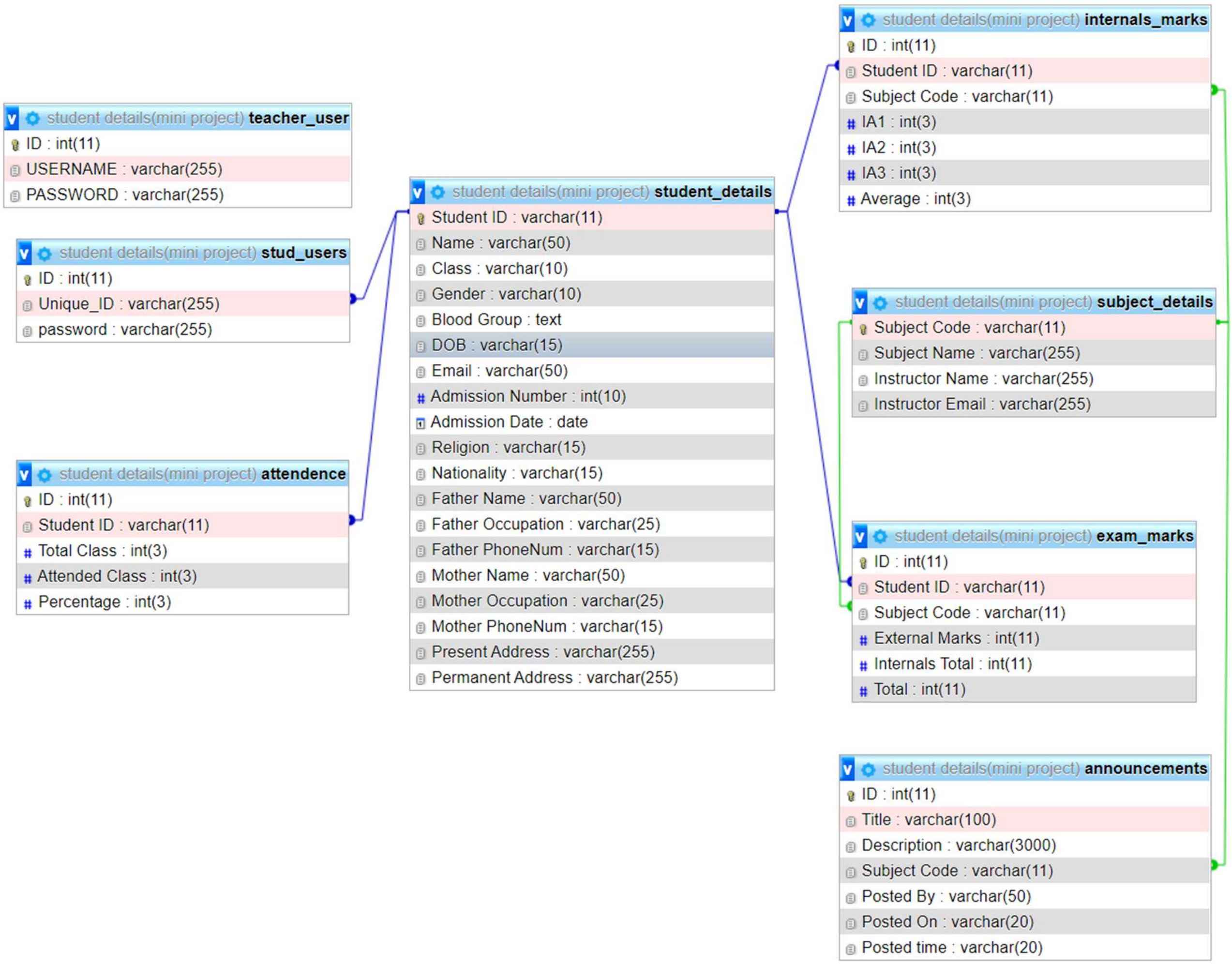
There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three-schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER-Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. The cardinality or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in order to explain how each aspect links together. In simple words Cardinality is a way to define the relationship between two entities.


Announcements
Posted time
Posted On
Posted By
Description
Title
ID
Announces
Subject Name
Instructor Name
Subject Code
Instructor Email
subject_details
internals_marks
ID
IA1
IA2
IA3
Average
SCORED_IN
SCORED_IN
exam_marks
Total
Internals Total
External Marks
ID
Secures
Secures
attendance
Percentage
Total Class
Attended Class
ID
Has
Student_Details
Name
Class
Gender
Blood Group
DOB
Email
Admission Number
Admission Date
Nationality
Student ID
Permanent Address
Present Address
Mother Name
Father Name
Religion
Father Occupation
Father PhoneNum
Mother PhoneNum
Mother Occupation
stud_user
password
ID
teacher_user
USERNAME
PASSWORD
ID

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute.
A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.



student details(mini project) **teacher_user**
ID : int(11)
USERNAME : varchar(255)
PASSWORD : varchar(255)

student details(mini project) **stud_users**
ID : int(11)
Unique_ID : varchar(255)
password : varchar(255)

student details(mini project) **attendence**
ID : int(11)
Student ID : varchar(11)
Total Class : int(3)
Attended Class : int(3)
Percentage : int(3)

student details(mini project) **student_details**
Student ID : varchar(11)
Name : varchar(50)
Class : varchar(10)
Gender : varchar(10)
Blood Group : text
DOB : varchar(15)
Email : varchar(50)
Admission Number : int(10)
Admission Date : date
Religion : varchar(15)
Nationality : varchar(15)
Father Name : varchar(50)
Father Occupation : varchar(25)
Father PhoneNum : varchar(15)
Mother Name : varchar(50)
Mother Occupation : varchar(25)
Mother PhoneNum : varchar(15)
Present Address : varchar(255)
Permanent Address : varchar(255)

student details(mini project) **internals_marks**
ID : int(11)
Student ID : varchar(11)
Subject Code : varchar(11)
IA1 : int(3)
IA2 : int(3)
IA3 : int(3)
Average : int(3)

student details(mini project) **subject_details**
Subject Code : varchar(11)
Subject Name : varchar(255)
Instructor Name : varchar(255)
Instructor Email : varchar(255)

student details(mini project) **exam_marks**
ID : int(11)
Student ID : varchar(11)
Subject Code : varchar(11)
External Marks : int(11)
Internals Total : int(11)
Total : int(11)

student details(mini project) **announcements**
ID : int(11)
Title : varchar(100)
Description : varchar(3000)
Subject Code : varchar(11)
Posted By : varchar(50)
Posted On : varchar(20)
Posted time : varchar(20)

3.4 DESCRIPTION OF TABLES

The database consists of 7 tables:

1) Teacher_user: Users (Teacher) login Details.
   - ID: Unique id, which is auto increment.
   - Username: Login User Name.
   - Password: Password associated with user to login into system.

2) Stud_user: Users (Student) login Details.
   - ID: Unique id, which is auto increment.
   - Unique_ID: Student ID for login.
   - Password: Password associated with user to login into system.

3) Attendance: Students Attendance Details
   - ID: Unique id, which is auto increment.
   - Student ID: Student ID (which is unique for each student).
   - Total Classes: Total Number of classes taken by Specific Subject Teacher.
   - Attended Classes: Total Classes attended by the Student.
   - Percentage: Percentage calculated using

     (Attended Classes / Total classes) * 100.

4) Student_details: Details of the Students in a specific class.
   - Student ID: Student ID (which is unique for each student).
   - Name: Name of the student.
   - Class: Class and Section, the pupil studying.
   - Gender: Gender of the Student.
   - Blood Group: Blood group of the Student.
   - DOB: Date of Birth of the Student.
   - Email: E-Mail ID of the Student.
   - Admission Number: Admission Number of the Student.
   - Admission Date: Admission Date of the Student.
   - Religion: Religion of the Student.
   - Nationality: Nationality of the Student.
   - Father Name: Student’s Father Name.
   - Father Occupation.
   - Father PhoneNum.
   - Mother Name: Student’s Mother Name.
   - Mother Occupation.
   - Mother PhoneNum.
   - Present Address: Present Residential Address.
   - Permanent Address: Permanent Residential Address.

5) Internals_marks: Formative Assessment Marks of a specific class.
   - ID: Unique id, which is auto increment.
   - Student ID: Student ID (which is unique for each student).
   - Subject Code: Subject Code (which is unique for each subject).
   - IA1: Marks of Internal Assessment - 1.
   - IA2: Marks of Internal Assessment - 2.
   - IA3: Marks of Internal Assessment - 3.
   - Average: Average of all the three Assessments.

6) Subject_details: Subject with respective Teacher.
   - Subject Code: Subject Code (which is unique for each subject).
   - Subject Name.
   - Instructor Name.
   - Instructor Email: Teacher’s E-Mail ID for Contact.

7) Exam_marks: Summative Assessment Marks of a specific class.
   - ID: Unique id, which is auto increment.
   - Student ID: Student ID (which is unique for each student).
   - Subject Code: Subject Code (which is unique for each subject).
   - Externals Marks: Exam marks for each subject.
   - Internals Total: Average of all the three Assessments.
   - Total: Sum of External and Internal Marks.

Chapter 4

IMPLEMENTATION

4.1 FUNCTIONAL MODULES

The functional modules included in the project are listed below:

- INSERT MODULE:

  This module provides the functionality of collecting the required data from the designed interface and transmitting it to the appropriate table present in the database designed for this project. If the provided data does not satisfy the given constraints, it must refrain from storing it into the database.

- UPDATE MODULE:

  This module again has the functionality of collecting the data from the designed interface, but it updates the already existing tuple that matches the provided primary key of the tuple to be updated, by replacing the existing attribute values with the newly collected data. Again, if the newly provided data does not satisfy the given constraints, it must refrain from updating the corresponding tuple.

- DELETE MODULE:

  The delete counterpart is loaded with the ability to delete a single or multiple record from the table. It searches for the tuple, in the query specified table, based on the provided value for an attribute. Admin can delete in the interface, based on which delete module searches for the record corresponding to that provided attribute value and deletes the record.

- RETRIEVE MODULE:

  The retrieve module has a basic functionality of accessing the entire specified table from the database and displays it.

- TRIGGER MODULE:

  Trigger in database is set of statements that are executed after an event occurs on the specified table. This is useful for logs wherein every change in database can be logged which helps keep a track of all changes/transactions on the database.

4.2 TABLE CREATION

- Table structure for Announcement:

```
CREATE TABLE `announcements` (
  `ID` int (11) AUTO_INCREMENT PRIMARY KEY,
  `Title` varchar (100) NOT NULL,
  `Description` varchar (3000) NOT NULL,
  `Subject Code` varchar (11) NOT NULL,
  `Posted By` varchar (50) NOT NULL,
  `Posted On` varchar (20) NOT NULL,
  `Posted time` varchar (20) NOT NULL,
  FOREIGN KEY (`Subject Code`) REFERENCES `subject_details` (`Subject Code`) ON
UPDATE CASCADE;
);
```

- Table structure for Attendence:

```
CREATE TABLE `attendence` (
  `ID` int (11) AUTO_INCREMENT PRIMARY KEY,
  `Student ID` varchar (11) DEFAULT NULL,
  `Total Class` int (3) DEFAULT NULL,
  `Attended Class` int (3) DEFAULT NULL,
  `Percentage` int (3) DEFAULT NULL,
  FOREIGN KEY (`Student ID`) REFERENCES `student_details` (`Student ID`) ON DELETE
CASCADE ON UPDATE CASCADE;
);
```

- Table structure for `Exam_marks`

```
CREATE TABLE `exam_marks` (
  `ID` int (11) AUTO_INCREMENT PRIMARY KEY,
  `Student ID` varchar (11) DEFAULT NULL,
  `Subject Code` varchar (11) DEFAULT NULL,
  `External Marks` int (11) DEFAULT NULL,
  `Internals Total` int (11) DEFAULT NULL,
  `Total` int (11) DEFAULT NULL,
  FOREIGN KEY (`Student ID`) REFERENCES `student_details` (`Student ID`) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Subject Code`) REFERENCES `subject_details` (`Subject Code`) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

- Table structure for table `Internals_marks`

```
CREATE TABLE `Internals_marks` (
  `ID` int (11)  AUTO_INCREMENT PRIMARY KEY,
  `Student ID` varchar (11) DEFAULT NULL,
  `Subject Code` varchar (11) DEFAULT NULL,
  `IA1` int (3) DEFAULT NULL,
  `IA2` int (3) DEFAULT NULL,
  `IA3` int (3) DEFAULT NULL,
  `Average` int (3) DEFAULT NULL,
  FOREIGN KEY (`Student ID`) REFERENCES `student_details` (`Student ID`) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Subject Code`) REFERENCES `subject_details` (`Subject Code`) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

- Table structure for `Student_details`

```
CREATE TABLE `student_details` (
  `Student ID` varchar (11) PRIMARY KEY,
  `Name` varchar (50) NOT NULL,
  `Class` varchar (10) NOT NULL,
  `Gender` varchar (10) NOT NULL,
  `Blood Group` text NOT NULL,
  `DOB` varchar (15) NOT NULL,
  `Email` varchar (50) NOT NULL,
  `Admission Number` int (10) NOT NULL,
  `Admission Date` date NOT NULL,
  `Religion` varchar (15) NOT NULL,
  `Nationality` varchar (15) NOT NULL,
  `Father Name` varchar (50) NOT NULL,
  `Father Occupation` varchar (25) NOT NULL,
  `Father PhoneNum` varchar (15) NOT NULL,
  `Mother Name` varchar (50) NOT NULL,
  `Mother Occupation` varchar (25) NOT NULL,
  `Mother PhoneNum` varchar (15) NOT NULL,
  `Present Address` varchar (255) NOT NULL,
  `Permanent Address` varchar (255) NOT NULL
);
```

- Table structure for `Stud_users`

```
CREATE TABLE `stud_users` (
 `ID` int (11) AUTO_INCREMENT PRIMARY KEY,
 `Unique_ID` varchar (255) NOT NULL,
 `password` varchar (255) DEFAULT NULL,
 FOREIGN KEY (`Unique_ID`) REFERENCES `student_details` (`Student ID`) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

- Table structure for `Subject_details`

```
CREATE TABLE `subject_details` (
 `Subject Code` varchar(11) PRIMARY KEY,
 `Subject Name` varchar(255) NOT NULL,
 `Instructor Name` varchar(255) NOT NULL,
 `Instructor Email` varchar(255) NOT NULL
);
```

- Table structure for table `Teacher_user`

```
CREATE TABLE `teacher_user` (
 `ID` int(11) AUTO_INCREMENT PRIMARY KEY,
 `USERNAME` varchar(255) NOT NULL,
 `PASSWORD` varchar(255) NOT NULL
);
```

4.3 TRIGGERS CREATION

- Triggers for `student_details` Table:

    - Insert Student ID into Attendence Table:

      ```
      CREATE TRIGGER `Add Student ID into attendance Table`
      AFTER INSERT ON `student_details`
      FOR EACH ROW INSERT INTO `attendence`
      (`Student ID`) VALUES (new.`Student ID`)
      ```

    - Insert Student ID into stud_users Table:

      ```
      CREATE TRIGGER `Add Student ID into Stud_users Table`
       AFTER INSERT ON `student_details`
      FOR EACH ROW INSERT INTO `stud_users`
      (`Unique_ID`) VALUES (new.`Student ID`)
      ```

    - Insert Student ID into exam_marks Table:

      ```
      CREATE TRIGGER `Add Student ID into exam_marks Table`
      AFTER INSERT ON `student_details`
      FOR EACH ROW INSERT INTO `exam_marks`
      (` Student ID `) VALUES (new.`Student ID`)
      ```

    - Insert Student ID into internals_marks Table:

      ```
      CREATE TRIGGER `Add Student ID into internals_marks Table`
      AFTER INSERT ON `student_details`
      FOR EACH ROW INSERT INTO `internals_marks`
      (`Student ID`) VALUES (new.`Student ID`)
      ```

- Triggers for `subject_details` Table:

    - Insert Subject Code into exam_marks Table:

      ```
      CREATE TRIGGER `Add Subject Code into exam_marks Table`
      AFTER INSERT ON `subject_details`
      FOR EACH ROW INSERT INTO `exam_marks`
      (` Subject Code`) VALUES (new.`Subject Code`)
      ```

    - Insert Subject Code into internals_marks Table:

      ```
      CREATE TRIGGER `Add Subject Code into internals_marks Table`
      AFTER INSERT ON `subject_details`
      FOR EACH ROW INSERT INTO `internals_marks`
      (` Subject Code`) VALUES (new.`Subject Code`)
      ```

4.4 RESULT

The resulting system is able to:

- **Teacher View:**
  - Register New Teacher credentials for login.
  - Login as Teacher for Academics records handling.
  - Admit, Update Students details for Specific Class.
  - View the Details of all the Students in the Class.
  - Add, update, delete the Subject details along with its instructor.
  - Create new Announcement and circulate the same.
  - Update Attendance for each student in the Class.
  - Update Internal Assessment Marks for each student.
  - Update Final Exam Marks For each Student.

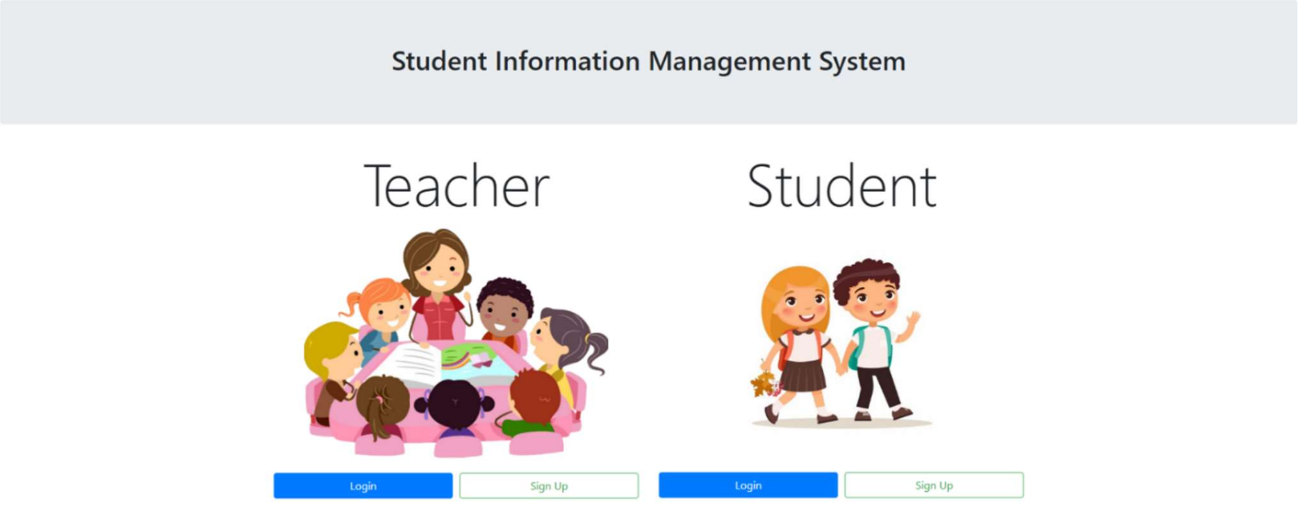
- **Student View:**
  - Register New Student credentials for login.
  - Login as Student for Curriculum Activities.
  - View any Recent Announcements made by specific Subject teacher.
  - View Subject Details and Contact information of the subject teachers.
  - View Internal Assessment Marks for each Subject.
  - View Final Exam Marks for each subject.
  - View the Attendance details.

Chapter 5

SNAPSHOTS

This chapter consists of working screenshots of the project.

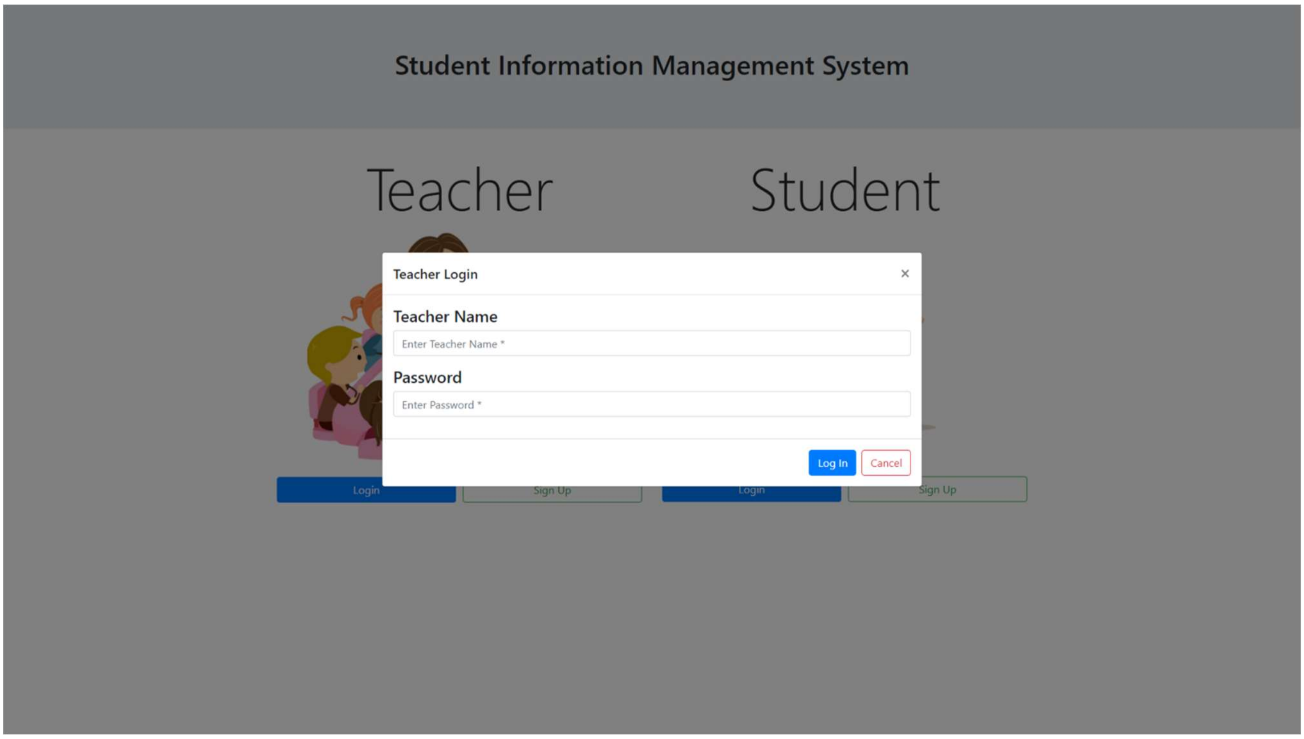
5.1 Welcome Page with Register and Login Page



Student Information Management System

Teacher

Student



Student Information Management System

Teacher

Student

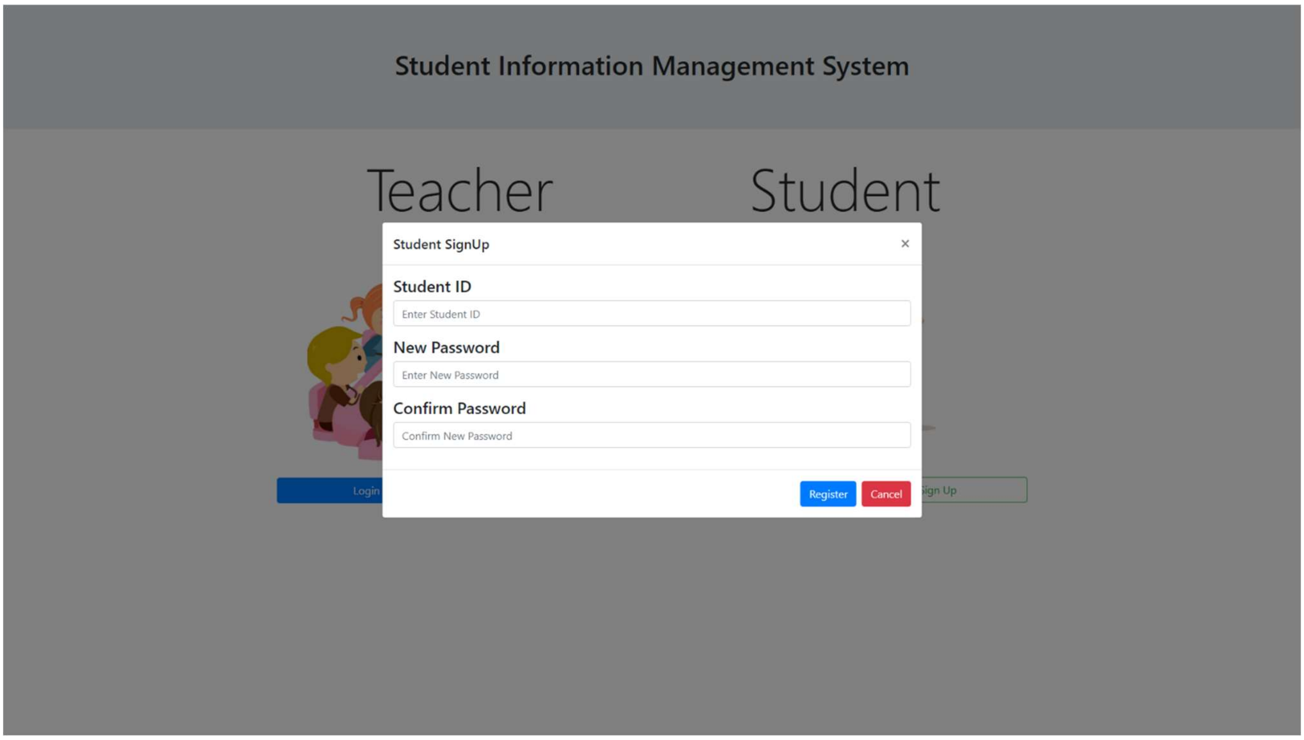
Teacher Login

Teacher Name

Enter Teacher Name *

Password

Enter Password *



Student Information Management System

Teacher Student

Student SignUp

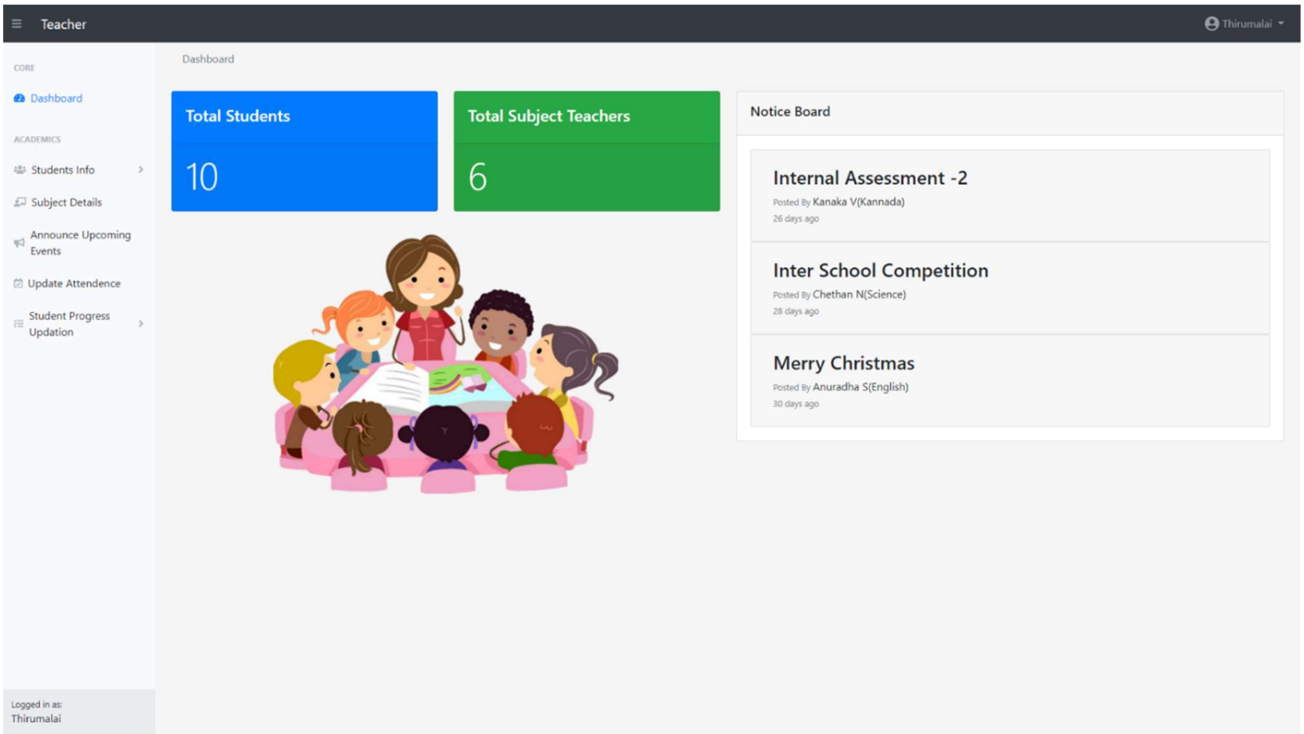
Student ID: Enter Student ID

New Password: Enter New Password

Confirm Password: Confirm New Password

5.2 Teacher View

5.2.1 Teacher Home Page



Total Students: 10

Total Subject Teachers: 6

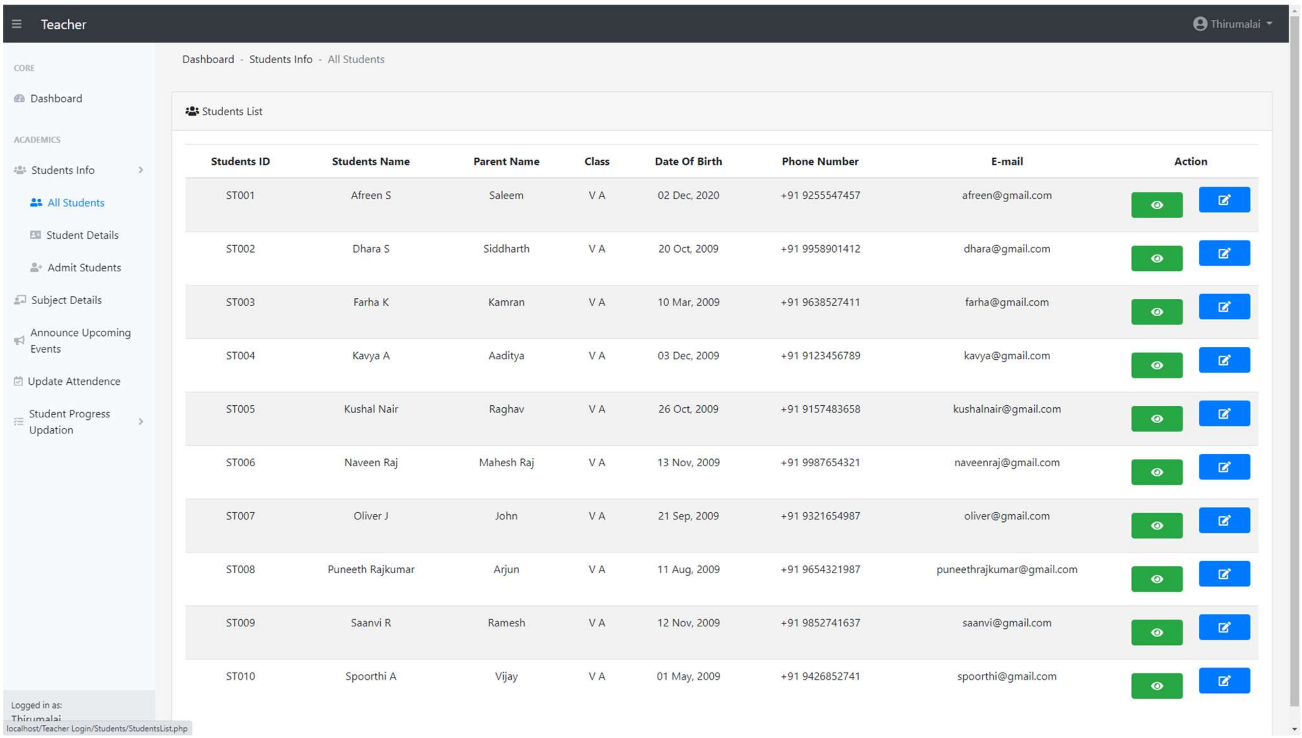
Notice Board

Internal Assessment -2
Posted By Kanaka V(Kannada)
26 days ago

Inter School Competition
Posted By Chethan N(Science)
28 days ago

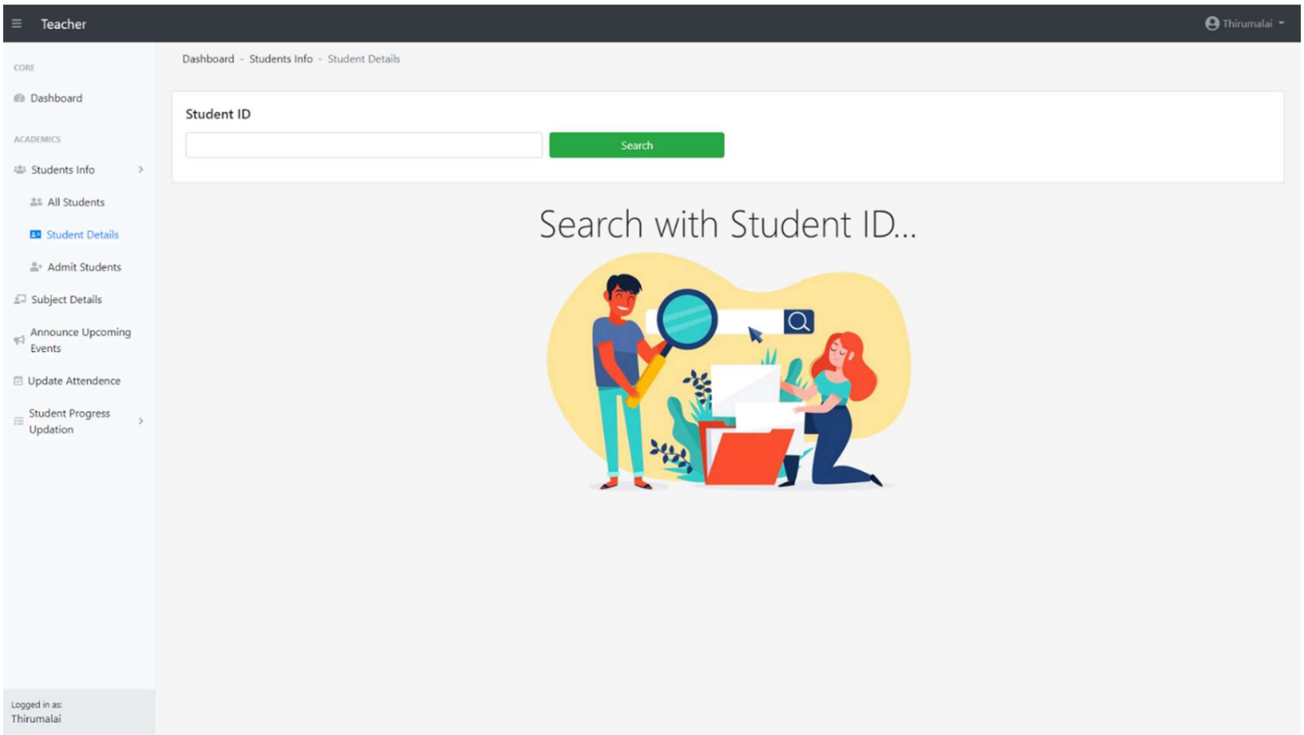
Merry Christmas
Posted By Anuradha S(English)
30 days ago

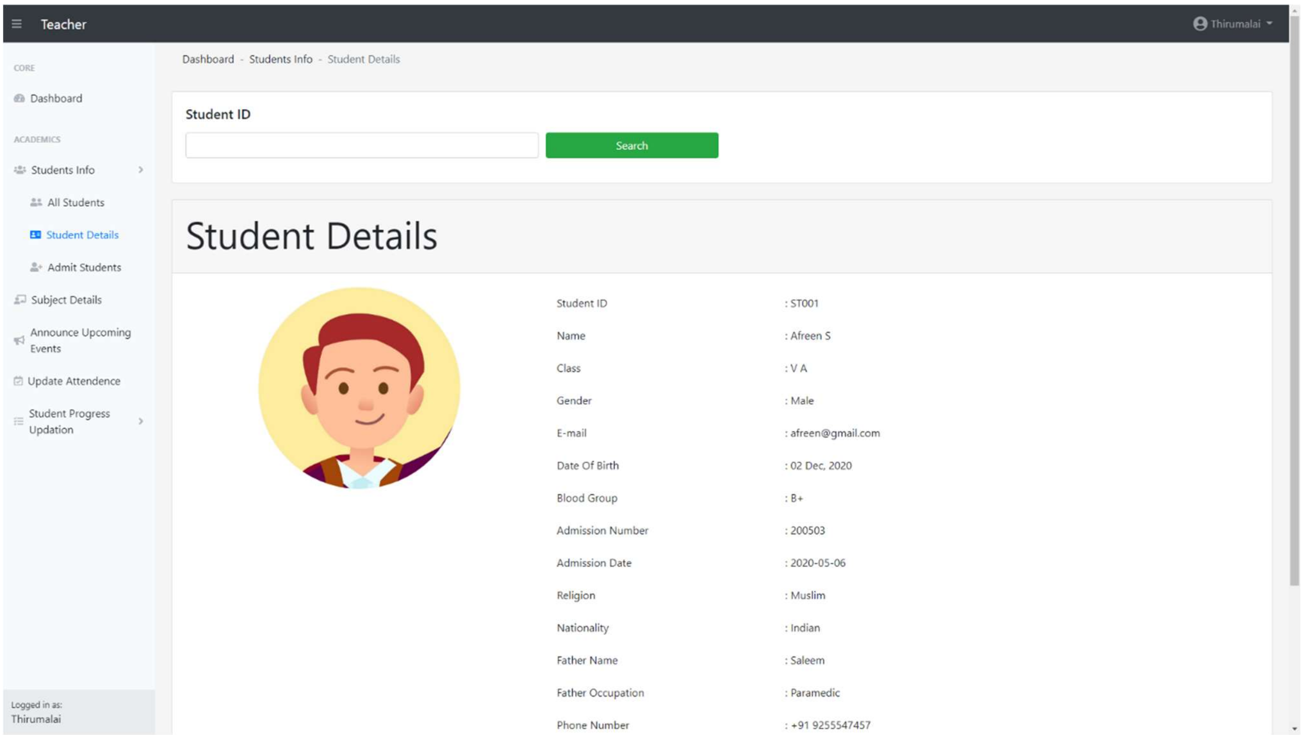
5.2.2 Admit, Update Students details for a Specific Class



Students ID	Students Name	Parent Name	Class	Date Of Birth	Phone Number	E-mail	Action
ST001	Afreen S	Saleem	V A	02 Dec, 2020	+91 9255547457	afreen@gmail.com	
ST002	Dhara S	Siddharth	V A	20 Oct, 2009	+91 9958901412	dhara@gmail.com	
ST003	Farha K	Kamran	V A	10 Mar, 2009	+91 9638527411	farha@gmail.com	
ST004	Kavya A	Aaditya	V A	03 Dec, 2009	+91 9123456789	kavya@gmail.com	
ST005	Kushal Nair	Raghav	V A	26 Oct, 2009	+91 9157483658	kushalnair@gmail.com	
ST006	Naveen Raj	Mahesh Raj	V A	13 Nov, 2009	+91 9987654321	naveenraj@gmail.com	
ST007	Oliver J	John	V A	21 Sep, 2009	+91 9321654987	oliver@gmail.com	
ST008	Puneeth Rajkumar	Arjun	V A	11 Aug, 2009	+91 9654321987	puneethrajkumar@gmail.com	
ST009	Saanvi R	Ramesh	V A	12 Nov, 2009	+91 9852741637	saanvi@gmail.com	
ST010	Spoorthi A	Vijay	V A	01 May, 2009	+91 9426852741	spoorthi@gmail.com	

5.2.3 Fetch or Search the Student Details Using Student ID



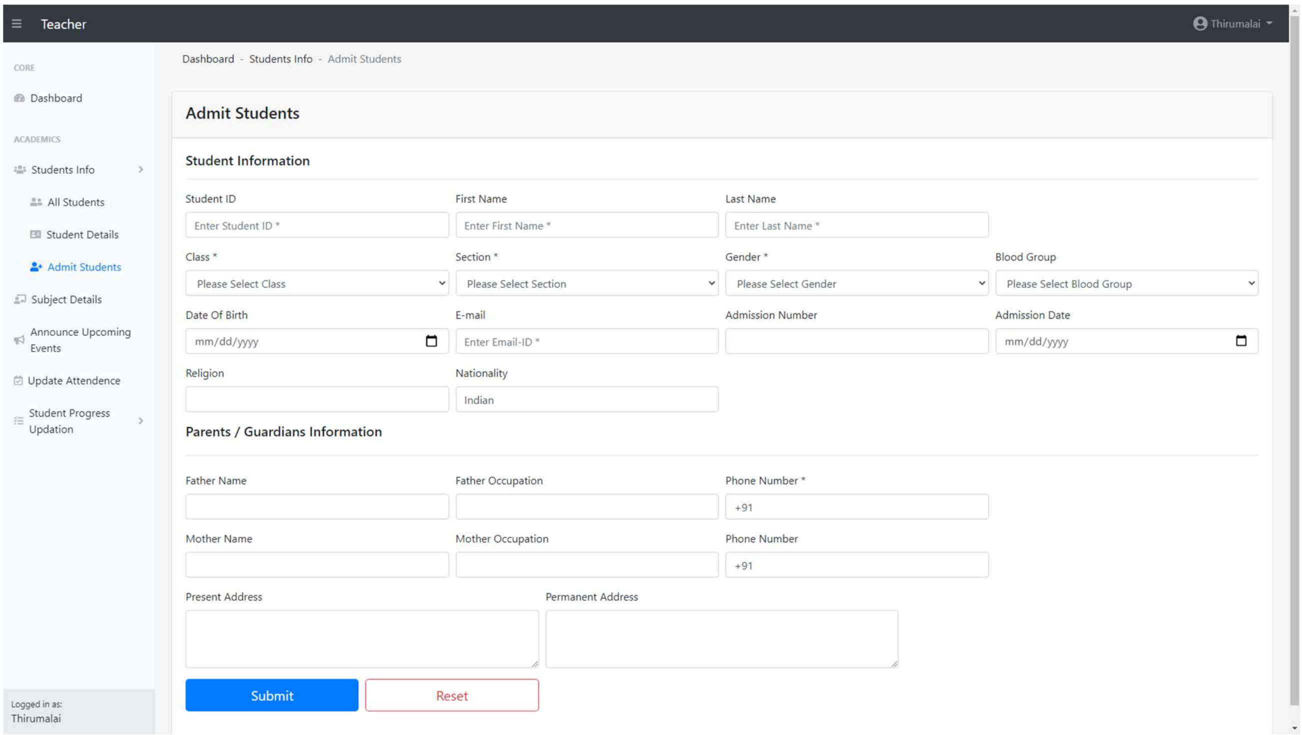


Student ID

Student Details

Student ID	: ST001
Name	: Afreen S
Class	: V A
Gender	: Male
E-mail	: afreen@gmail.com
Date Of Birth	: 02 Dec, 2020
Blood Group	: B+
Admission Number	: 200503
Admission Date	: 2020-05-06
Religion	: Muslim
Nationality	: Indian
Father Name	: Saleem
Father Occupation	: Paramedic
Phone Number	: +91 9255547457

5.2.4 Admit or Add new Student Details to the Specific Class



Admit Students

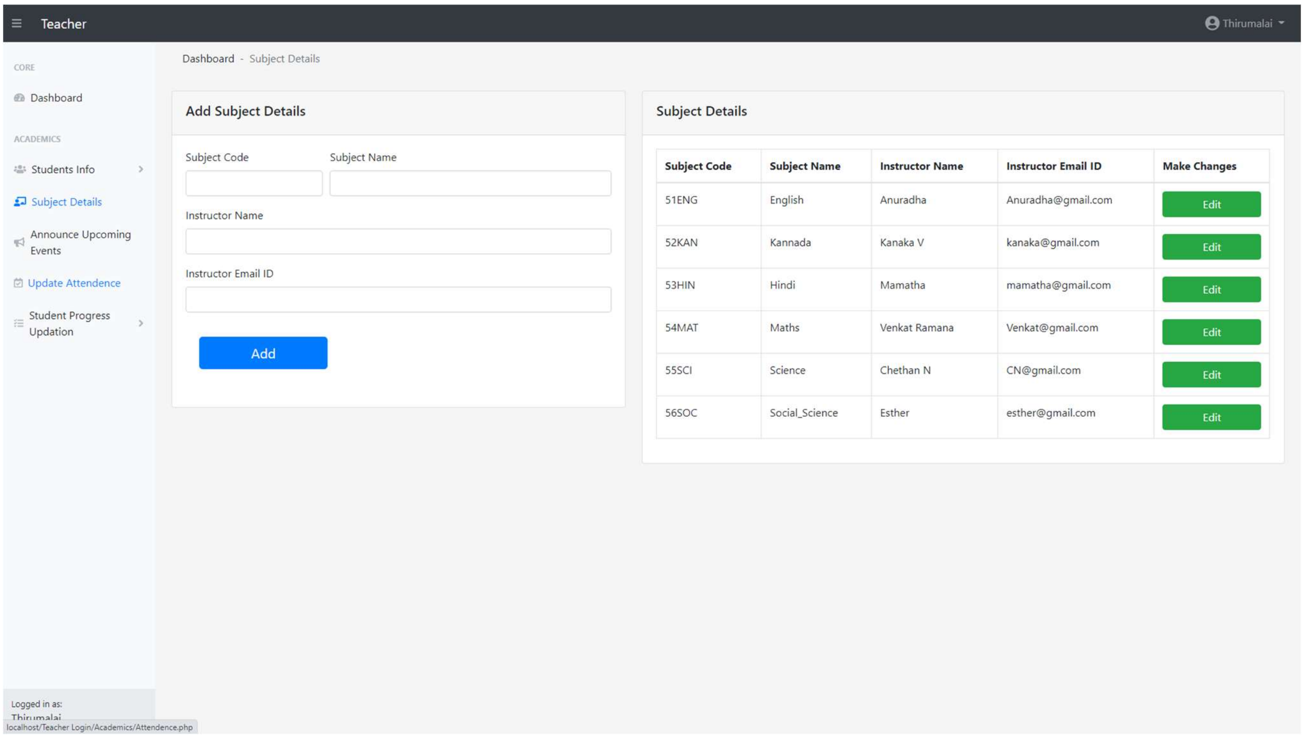
Student Information

Student ID: Enter Student ID *
First Name: Enter First Name *
Last Name: Enter Last Name *
Class *: Please Select Class
Section *: Please Select Section
Gender *: Please Select Gender
Blood Group: Please Select Blood Group
Date Of Birth: mm/dd/yyyy
E-mail: Enter Email-ID *
Admission Number:
Admission Date: mm/dd/yyyy
Religion:
Nationality: Indian

Parents / Guardians Information

Father Name:
Father Occupation:
Phone Number *: +91
Mother Name:
Mother Occupation:
Phone Number: +91
Present Address:
Permanent Address:

5.2.5 Add or Update Subject with its Instructor Details



Add Subject Details

Subject Code

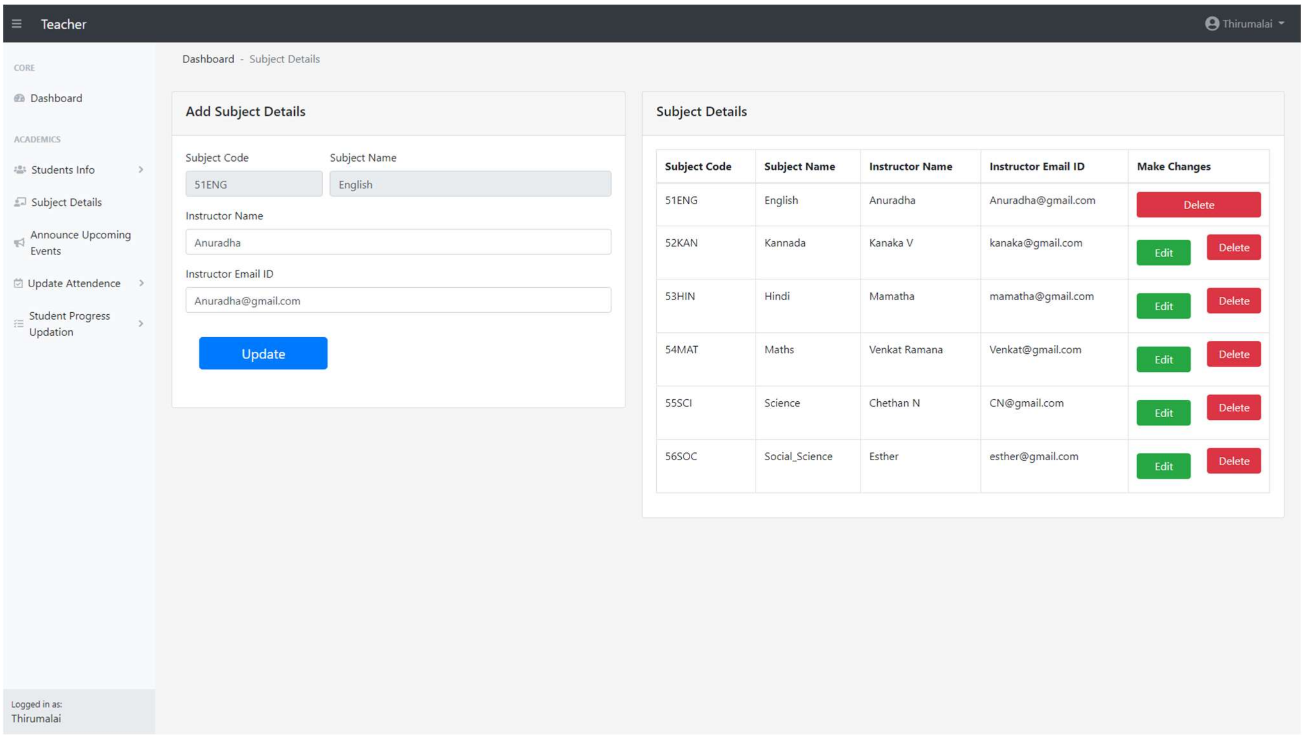
Subject Name

Instructor Name

Instructor Email ID

Subject Details

Subject Code	Subject Name	Instructor Name	Instructor Email ID	Make Changes
51ENG	English	Anuradha	Anuradha@gmail.com	Edit
52KAN	Kannada	Kanaka V	kanaka@gmail.com	Edit
53HIN	Hindi	Mamatha	mamatha@gmail.com	Edit
54MAT	Maths	Venkat Ramana	Venkat@gmail.com	Edit
55SCI	Science	Chethan N	CN@gmail.com	Edit
56SOC	Social_Science	Esther	esther@gmail.com	Edit



Add Subject Details

Subject Code: 51ENG

Subject Name: English

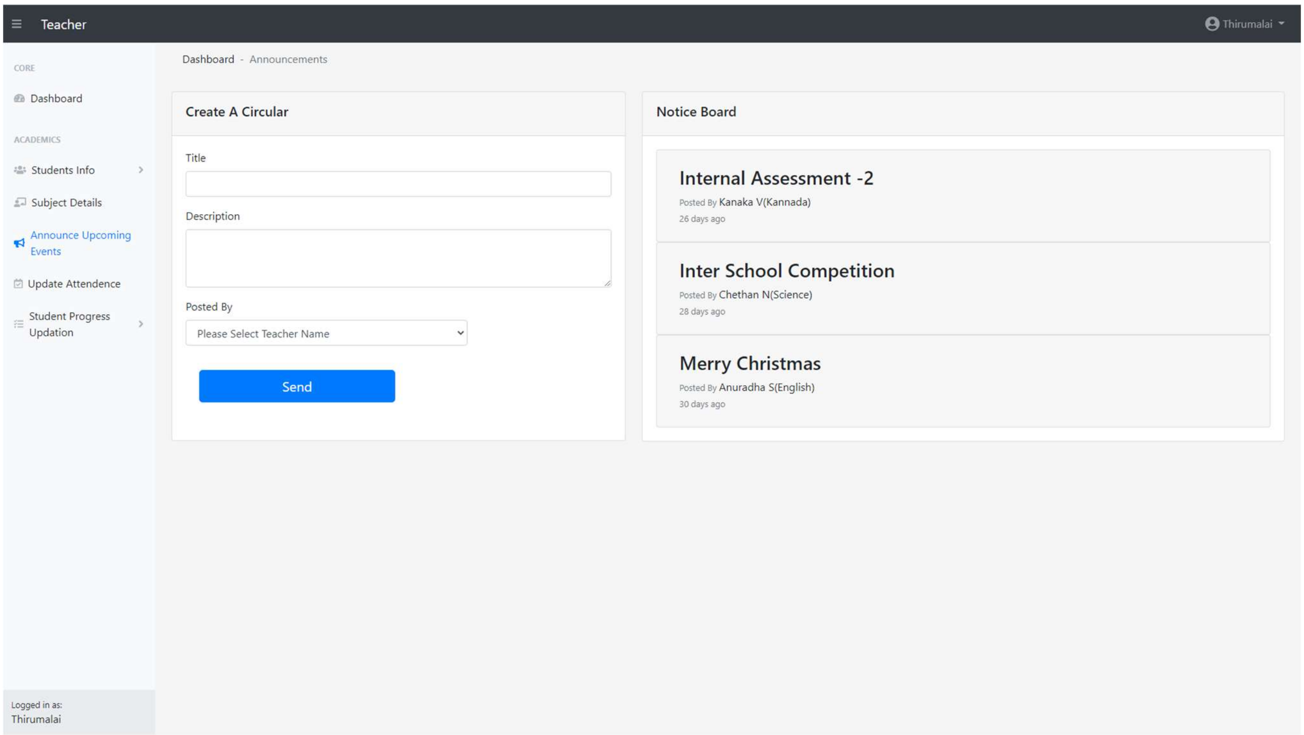
Instructor Name: Anuradha

Instructor Email ID: Anuradha@gmail.com

Subject Details

Subject Code	Subject Name	Instructor Name	Instructor Email ID	Make Changes
51ENG	English	Anuradha	Anuradha@gmail.com	Delete
52KAN	Kannada	Kanaka V	kanaka@gmail.com	Edit Delete
53HIN	Hindi	Mamatha	mamatha@gmail.com	Edit Delete
54MAT	Maths	Venkat Ramana	Venkat@gmail.com	Edit Delete
55SCI	Science	Chethan N	CN@gmail.com	Edit Delete
56SOC	Social_Science	Esther	esther@gmail.com	Edit Delete

5.2.6 Announce Upcoming Events and Circulate the Same



Create A Circular

Title

Description

Posted By

Please Select Teacher Name

Send

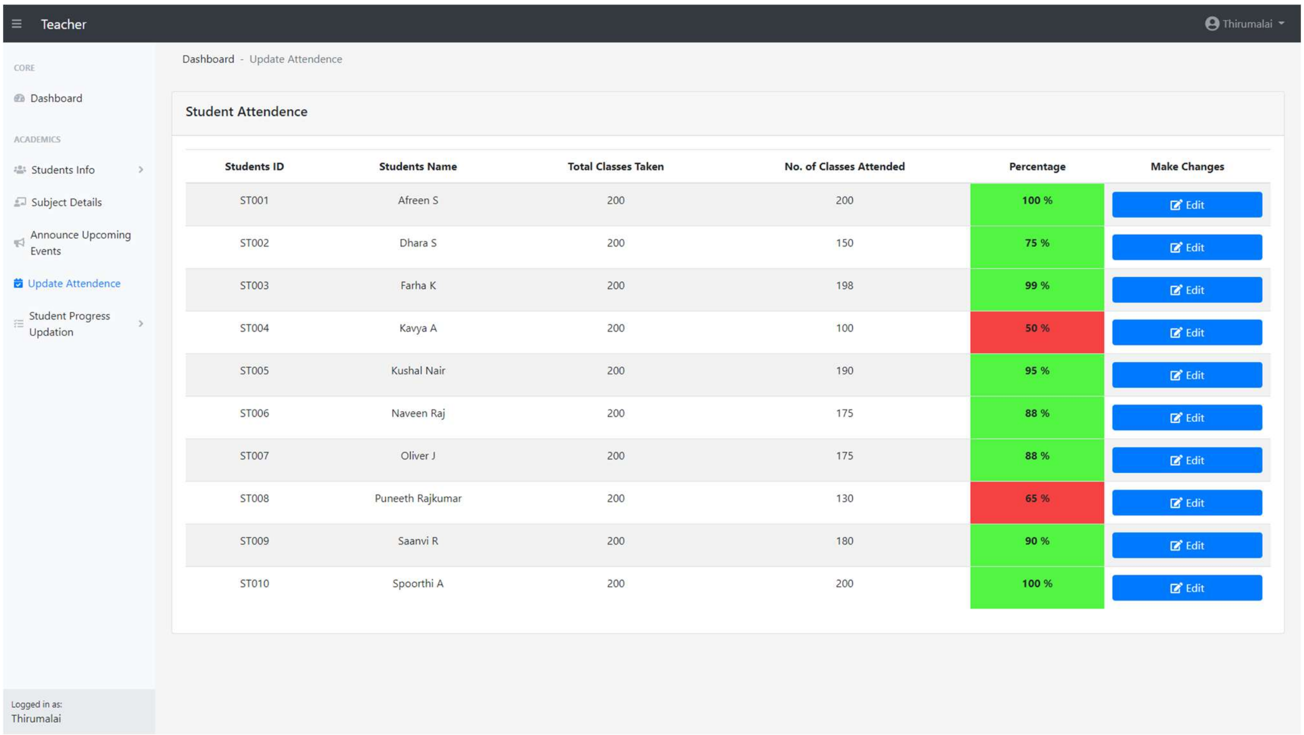
Notice Board

Internal Assessment -2
Posted By Kanaka V(Kannada)
26 days ago

Inter School Competition
Posted By Chethan N(Science)
28 days ago

Merry Christmas
Posted By Anuradha S(English)
30 days ago

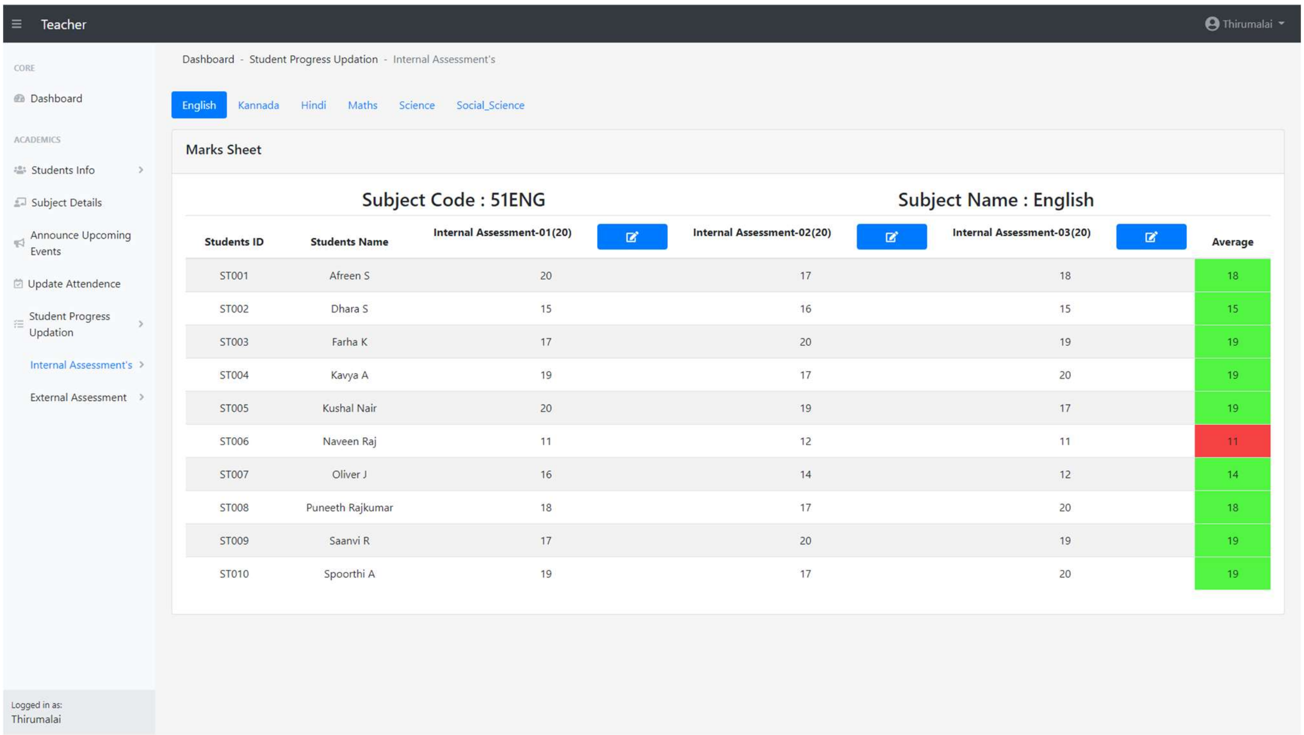
5.2.7 Update Attendance for each student in the Class



Student Attendence

Students ID	Students Name	Total Classes Taken	No. of Classes Attended	Percentage	Make Changes
ST001	Afreen S	200	200	100 %	Edit
ST002	Dhara S	200	150	75 %	Edit
ST003	Farha K	200	198	99 %	Edit
ST004	Kavya A	200	100	50 %	Edit
ST005	Kushal Nair	200	190	95 %	Edit
ST006	Naveen Raj	200	175	88 %	Edit
ST007	Oliver J	200	175	88 %	Edit
ST008	Puneeth Rajkumar	200	130	65 %	Edit
ST009	Saanvi R	200	180	90 %	Edit
ST010	Spoorthi A	200	200	100 %	Edit

5.2.8 Update Internal Assessment Marks for each student

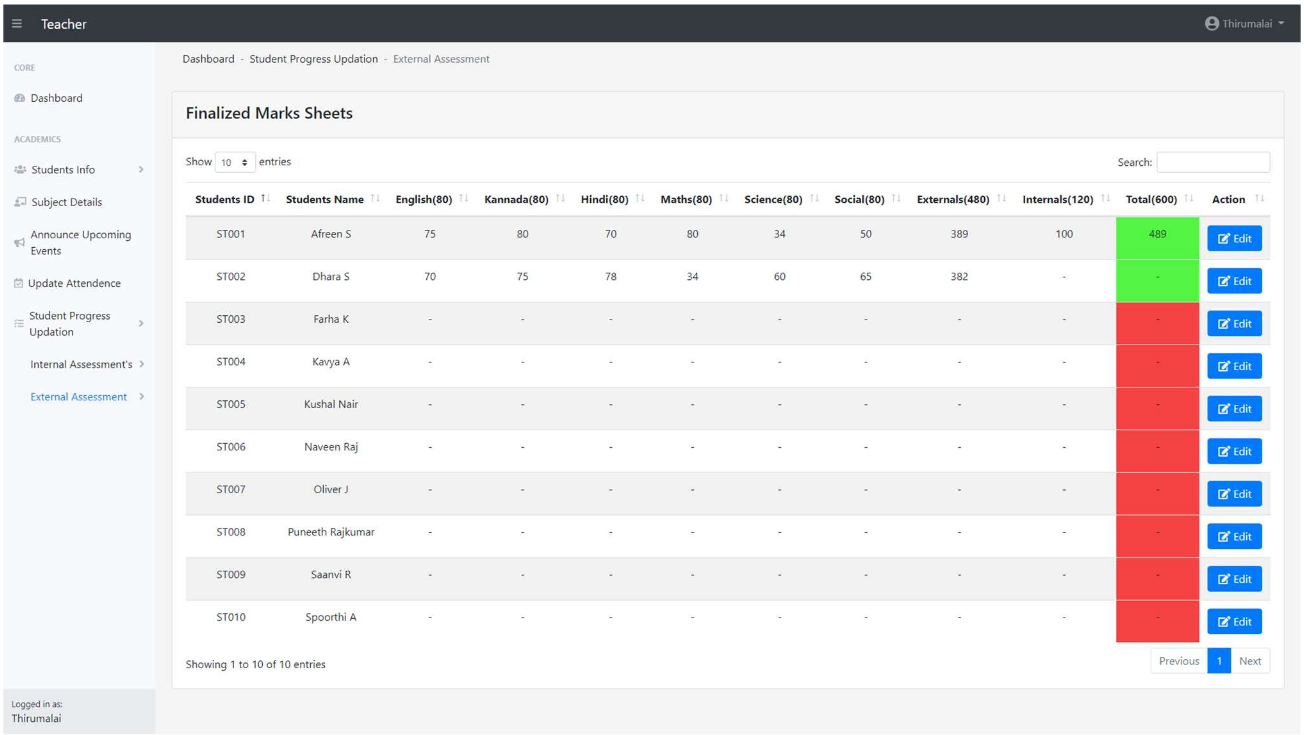


Marks Sheet

Subject Code : 51ENG Subject Name : English

Students ID	Students Name	Internal Assessment-01(20)	Internal Assessment-02(20)	Internal Assessment-03(20)	Average
ST001	Afreen S	20	17	18	18
ST002	Dhara S	15	16	15	15
ST003	Farha K	17	20	19	19
ST004	Kavya A	19	17	20	19
ST005	Kushal Nair	20	19	17	19
ST006	Naveen Raj	11	12	11	11
ST007	Oliver J	16	14	12	14
ST008	Puneeth Rajkumar	18	17	20	18
ST009	Saanvi R	17	20	19	19
ST010	Spoorthi A	19	17	20	19

5.2.9 Update Exam Marks for each student



Finalized Marks Sheets

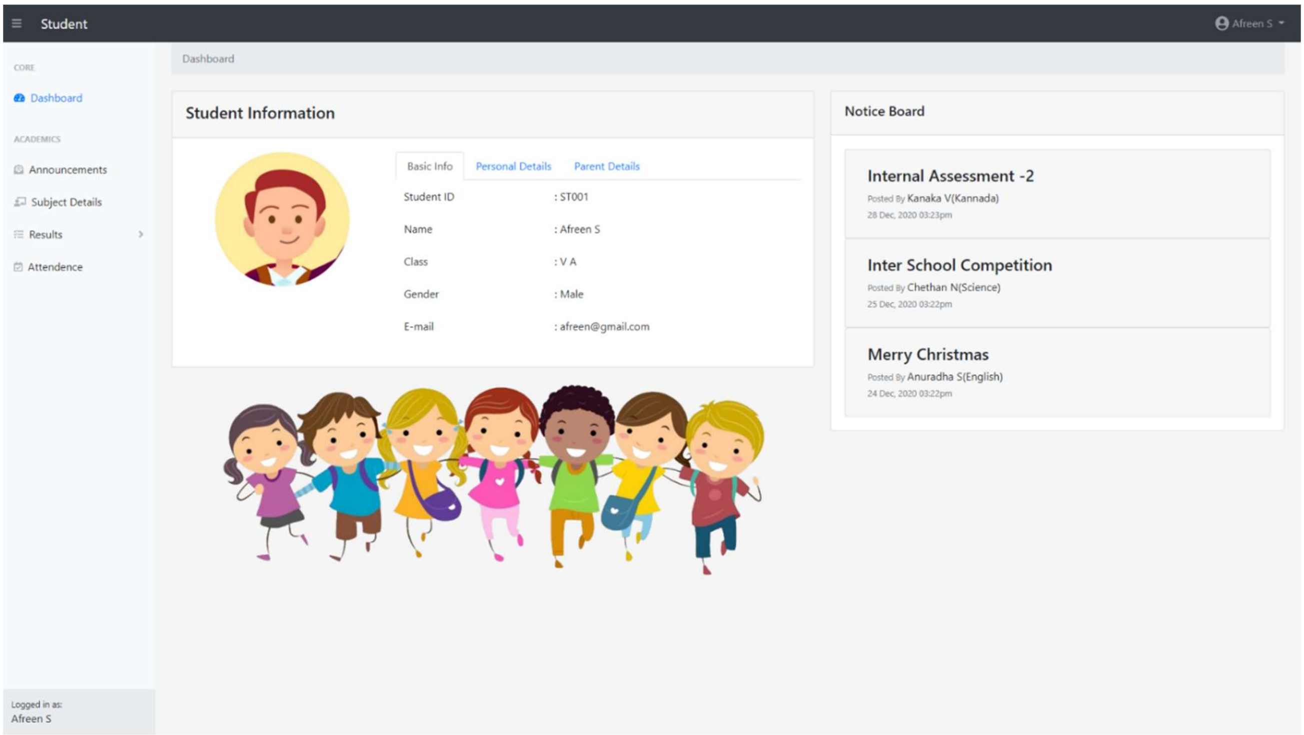
Show 10 entries Search:

Students ID	Students Name	English(80)	Kannada(80)	Hindi(80)	Maths(80)	Science(80)	Social(80)	Externals(480)	Internals(120)	Total(600)	Action
ST001	Afreen S	75	80	70	80	34	50	389	100	489	Edit
ST002	Dhara S	70	75	78	34	60	65	382	-	-	Edit
ST003	Farha K	-	-	-	-	-	-	-	-	-	Edit
ST004	Kavya A	-	-	-	-	-	-	-	-	-	Edit
ST005	Kushal Nair	-	-	-	-	-	-	-	-	-	Edit
ST006	Naveen Raj	-	-	-	-	-	-	-	-	-	Edit
ST007	Oliver J	-	-	-	-	-	-	-	-	-	Edit
ST008	Puneeth Rajkumar	-	-	-	-	-	-	-	-	-	Edit
ST009	Saanvi R	-	-	-	-	-	-	-	-	-	Edit
ST010	Spoorthi A	-	-	-	-	-	-	-	-	-	Edit

Showing 1 to 10 of 10 entries

5.3 Student View

5.3.1 Student Home Page



Student Information

Student ID : ST001

Name : Afreen S

Class : V A

Gender : Male

E-mail : afreen@gmail.com

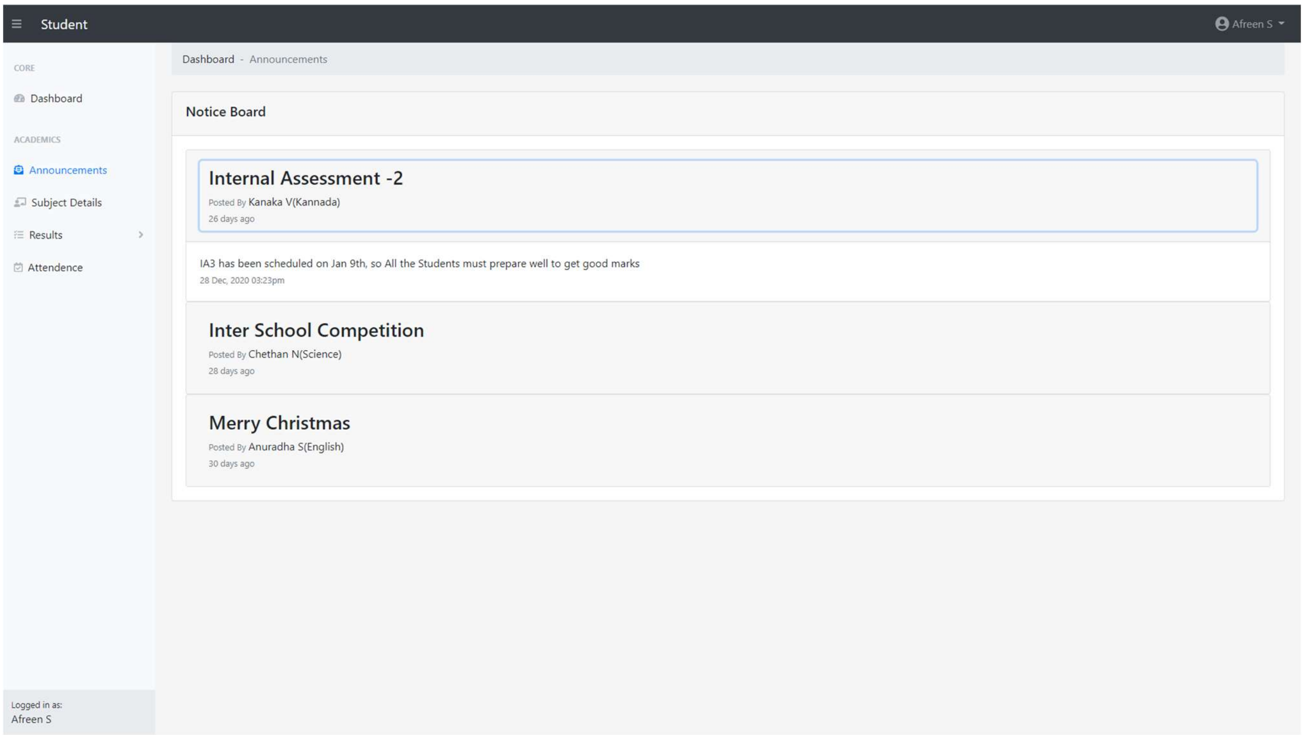
Notice Board

Internal Assessment -2
Posted By Kanaka V(Kannada)
28 Dec, 2020 03:23pm

Inter School Competition
Posted By Chethan N(Science)
25 Dec, 2020 03:22pm

Merry Christmas
Posted By Anuradha S(English)
24 Dec, 2020 03:22pm

5.3.2 View any Recent Announcements made by specific Subject teacher



Notice Board

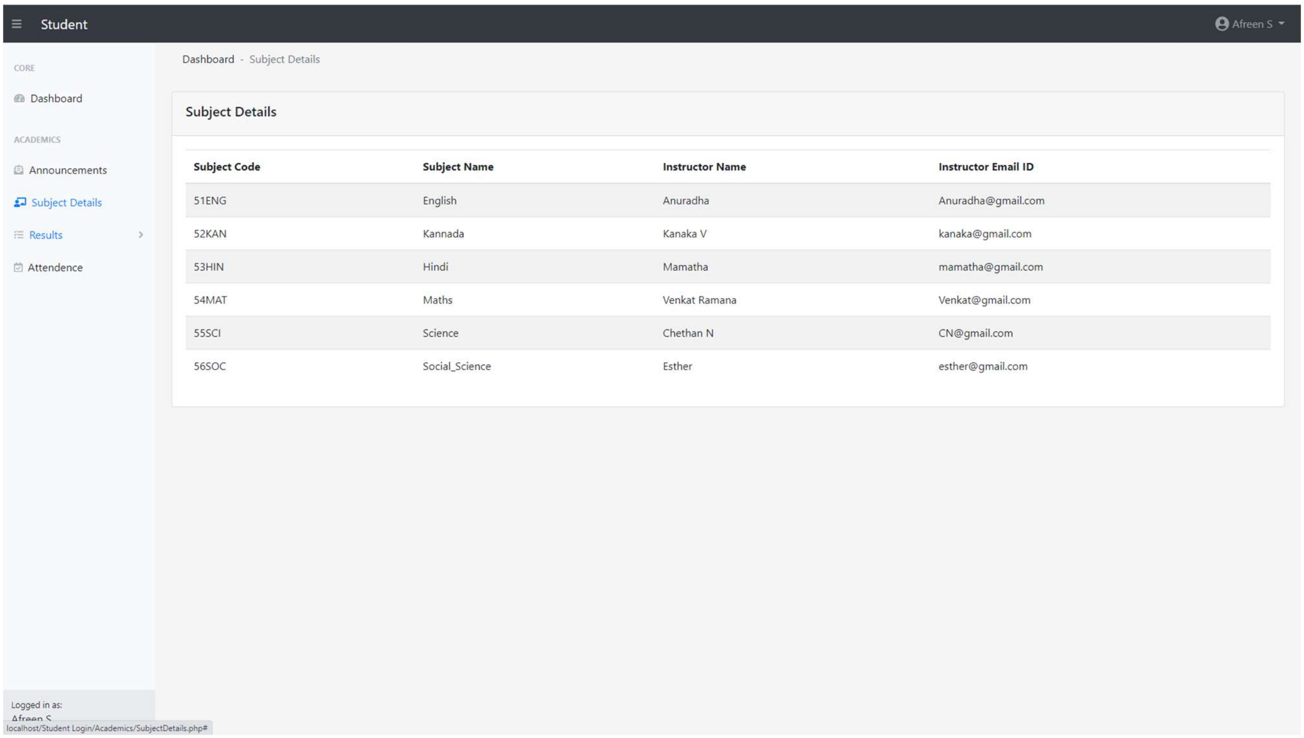
Internal Assessment -2
Posted By Kanaka V(Kannada)
26 days ago

IA3 has been scheduled on Jan 9th, so All the Students must prepare well to get good marks
28 Dec, 2020 03:23pm

Inter School Competition
Posted By Chethan N(Science)
28 days ago

Merry Christmas
Posted By Anuradha S(English)
30 days ago

5.3.3 View Subject Details and Contact information of the subject teachers

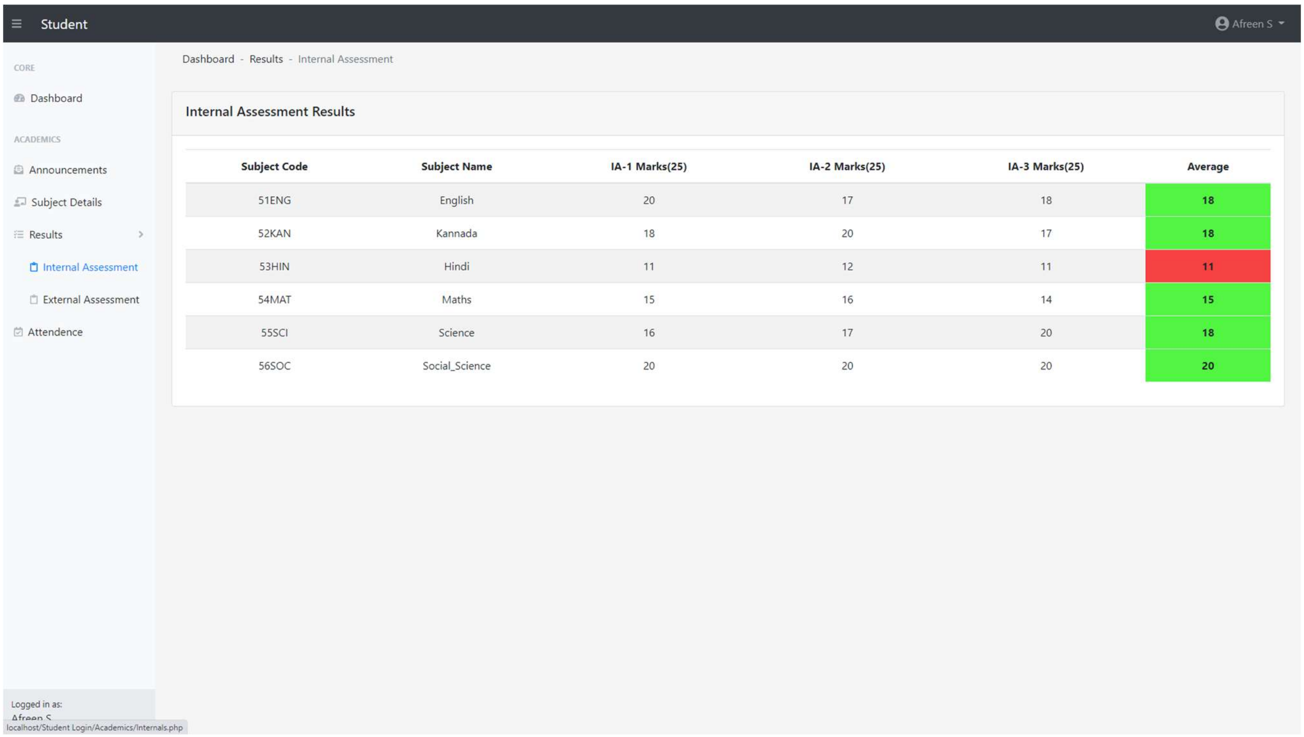


Dashboard · Subject Details

Subject Details

Subject Code	Subject Name	Instructor Name	Instructor Email ID
51ENG	English	Anuradha	Anuradha@gmail.com
52KAN	Kannada	Kanaka V	kanaka@gmail.com
53HIN	Hindi	Mamatha	mamatha@gmail.com
54MAT	Maths	Venkat Ramana	Venkat@gmail.com
55SCI	Science	Chethan N	CN@gmail.com
56SOC	Social_Science	Esther	esther@gmail.com

5.3.4 View Internal Assessment Marks for each Subject

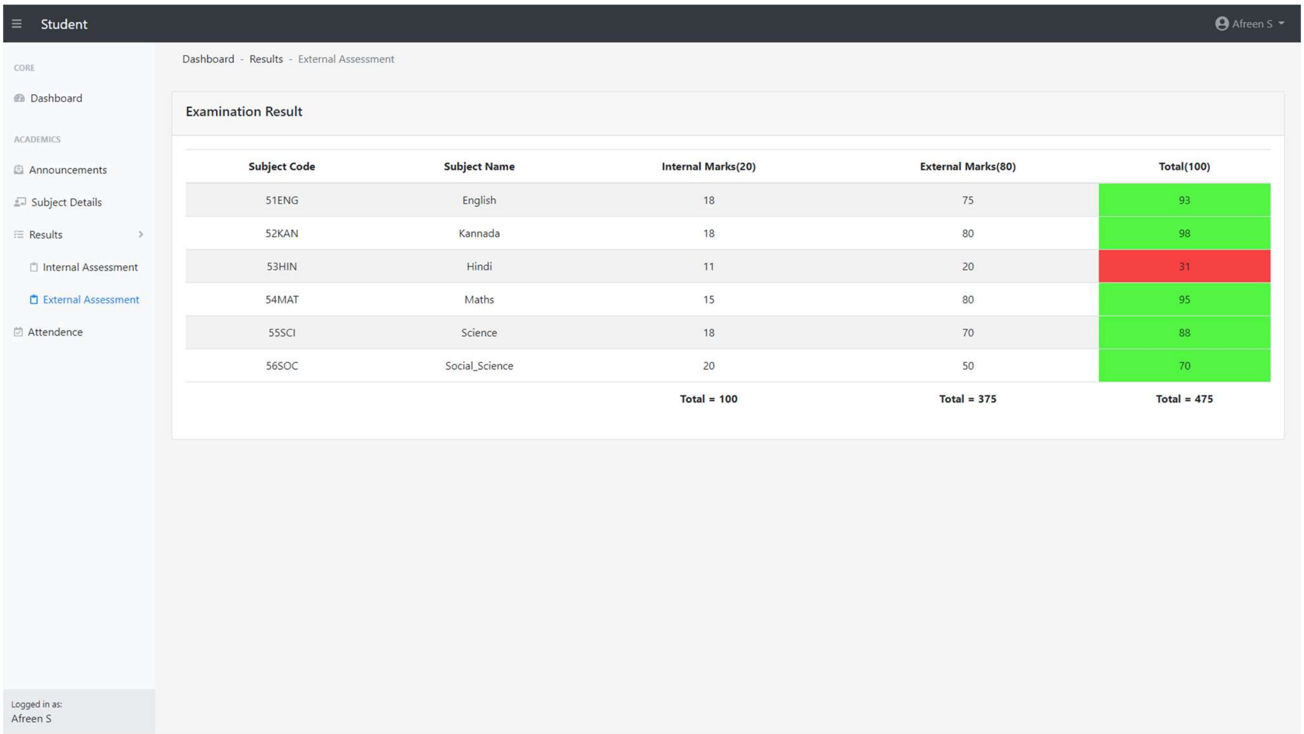


Dashboard · Results · Internal Assessment

Internal Assessment Results

Subject Code	Subject Name	IA-1 Marks(25)	IA-2 Marks(25)	IA-3 Marks(25)	Average
51ENG	English	20	17	18	18
52KAN	Kannada	18	20	17	18
53HIN	Hindi	11	12	11	11
54MAT	Maths	15	16	14	15
55SCI	Science	16	17	20	18
56SOC	Social_Science	20	20	20	20

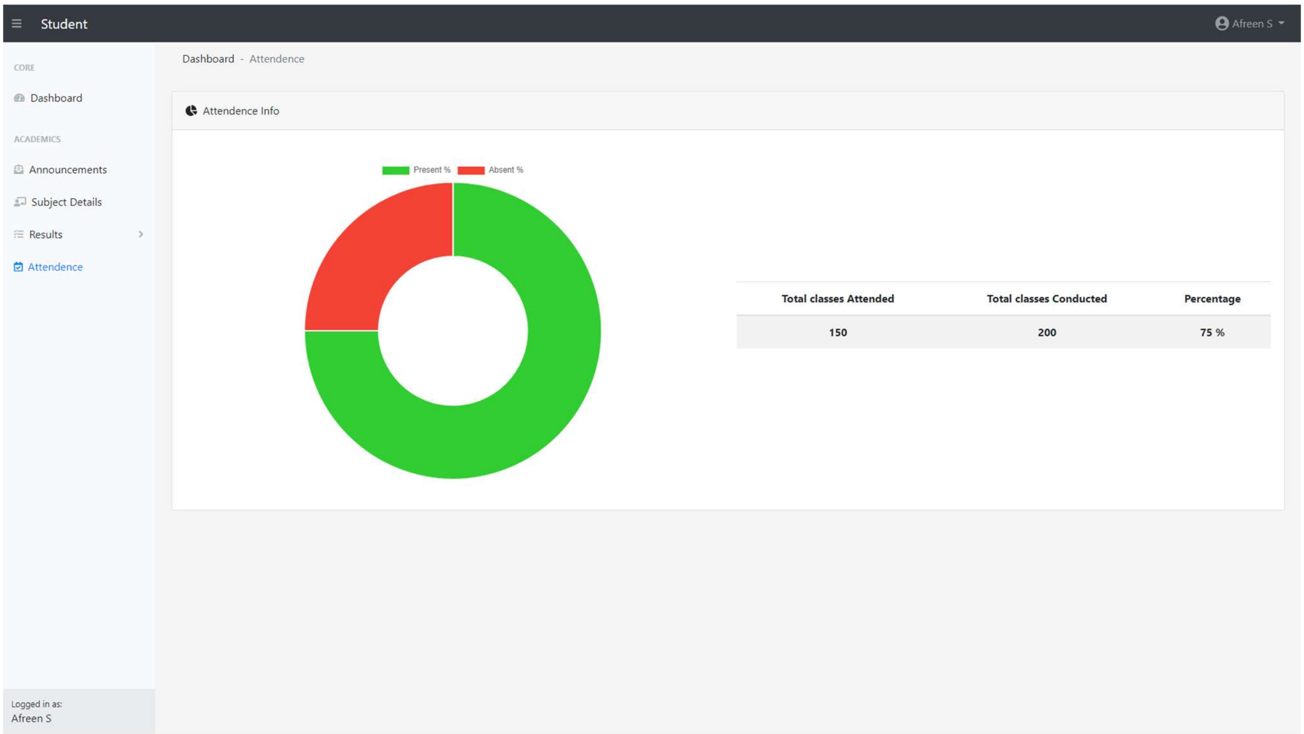
5.3.5 View Final Exam Marks for each subject



Examination Result

Subject Code	Subject Name	Internal Marks(20)	External Marks(80)	Total(100)
51ENG	English	18	75	93
52KAN	Kannada	18	80	98
53HIN	Hindi	11	20	31
54MAT	Maths	15	80	95
55SCI	Science	18	70	88
56SOC	Social_Science	20	50	70
		Total = 100	Total = 375	Total = 475

5.3.6 View the Attendance details



Attendence Info

Total classes Attended	Total classes Conducted	Percentage
150	200	75 %

Chapter 6

TESTING

6.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

6.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

6.3 LIMITATIONS

- Our site does reduce major paper work, but still need some manual data recording like admission, exam data, etc.
- Our site is not fully integrated to fulfil every activities of the student.
- Does not support inbuilt communication between parents and teachers.

CONCLUSION

The purpose of implementing the 'Student Information Management System' is for the ease in maintaining student related data on an institution, our system does serve this purpose and also is a means of communication between the students/parents and teachers during times when physical contact is not possible or when it is necessary to convey an important message quickly, due to this system a lot of paper work, maintenance, space, time are all reduced for the purpose of handling large student data.

The goals achieved by this project are:

- Centralized Database.
- Reduced paper works and Stacking of files.
- Easier Searching, Updating and Storing the Details
- User Friendly Environment.

FUTURE ENHANCEMENTS

Along with the existing features, additional features can be added in the future, Like:

- Handling admission details in a more detailed manner by Adding payment feature through our site.
- Extension of the current system for the entire institution.
- Better interaction between the parents and the teachers.
- Storing the records of co-curricular activities.
- Assigning and reporting the assignments and many more.

REFERENCES

- https://www.w3schools.com
- https://getbootstrap.com/
- https://github.com/ (open source)
- https://www.youtube.com/c/ThapaTechnical/playlists
- https://fontawesome.com
- https://stackoverflow.com/
- https://www.google.com/