



INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

भारतीय प्रौद्योगिकी संस्थान तिरुपति

College ERP System

Mini project report for the Industrial Software Engineering (CS5206)

Submitted By: Susanta Ghosh (CS18M014)

Under the Guidance of

Dr.Sridhar Chimalakonda

Assistant Professor

Dept.of Computer Science & Engineering

Abstract

This report specifies the various processes and techniques used in gathering requirements, designing, implementing and testing for the project on the college management system. The problems regarding the current system in the college were analyzed and noted. This project aims to solve some of those problems and thus, add more value to the current system. The requirements were gathered from all the stakeholders and based on that we created a requirements model and designed the software based on the based. The project was implemented in the form of a website using django(python).

Using the various resources and tools we gathered along the way, we implemented the college ERP system using some features that solve the current problems in the system such as a provision to edit the attendance and marks before locking it at the end. The software was also tested using the various testing methods and results were positive. Thus, the results can be integrated in the current ERP system to improve it's working and solve some of the existing problems.

Introduction

The objective of the College Information Management System is to allow the administrator of any organization the ability to edit and find out the personal details of a student and allows the student to keep up to date his profile. It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about a student will be available in a few seconds. Overall, it'll make Student Information an easier job for the administrator and the student of any organization.

The main purpose of this project is to illustrate the requirements of the project College Information Management System and is intended to help any organization to maintain and manage personal data. It is a comprehensive project developed from the ground up to fulfill the needs of colleges as they guide their students. This integrated information management system connects daily

operations in the college environment ranging from Attendance management to communicational means among students and teachers. This reduces data error and ensures that information is always up-to-date throughout the college. It provides a single source of data repository for streamlining your processes and for all reporting purposes. It has a simple user interface and is intuitive. This ensures that the users spend less time in learning the system and hence, increase their productivity. Efficient security features provide data privacy and hence, increase their productivity.

- **Introduction to problem domain**

As we know, a college consists of different departments, such as course departments, fees management, library, event management etc. Nowadays applications and uses of information technologies are increasing as compared to before, each of these individual departments has its own computer system to do their own functionalities. By having one main system they can interact with each other from their respected system by having valid user id and password.

Aim of the problem

The objective of the College Information Management System is to allow the administrator of any organization the ability to edit and find out the personal details of a student and allows the student to keep up to date his profile. It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about a student will be available in a few seconds. Overall, it'll make Student Information an easier job for the administrator and the student of any organization.

The main purpose of this project is to illustrate the requirements of the project College Information Management System and is intended to help any organization to maintain and manage personal data. It is a comprehensive

project developed from the ground up to fulfill the needs of colleges as they guide their students. This integrated information management system connects daily operations in the college environment ranging from Attendance management to communicational means among students and teachers. This reduces data error and ensures that information is always up-to-date throughout the college. It provides a single source of data repository for streamlining your processes and for all reporting purposes. It has a simple user interface and is intuitive. This ensures that the users spend less time in learning the system and hence, increase their productivity. Efficient security features provide data privacy and hence, increase their productivity.

System Requirement Engineering

- **Inception**

Inception is a process of establishing a basic understanding of the problem and the nature of the solution. This includes the need for this software, identification of stakeholders and defining multiple viewpoints.

What is the purpose of this project?

There is currently an ERP system in our college. But, not everyone is happy with the system. While it is a step towards automating the college activities, it comes with its own set of problems. This project is designed to implement a college ERP system to eradicate some of these problems and add some features of our own that would add value to the system.

Why do we need ERP?

Nowadays, in schools and colleges, it is very difficult to manage each and everything manually. Supervising and maintaining the whole database of a school or college can be time-consuming and challenging especially if it's done on a regular basis. So, we need to handle and manage everything smartly. To solve this problem ERP(Enterprise Resource Planning) is used. ERP software makes it easy to track the progress of every department of school and automate different functions. With ERP everything can be seen on a single dashboard. The administrator can manage the college from anywhere. The possibilities of maintaining the whole database of a college with ERP software are endless. Some of the prominent roles of ERP are:

- Manages the office and automates different functions.
- Helps in long-term management and planning of all departments of college.
- Eliminates the need for having multiple management software for each department.
- Daily activities like attendance can be digitized and automated.
- Leave module for teachers can be automated.

● **Identification of stakeholders**

Enterprise Resource planning implementation is a difficult and complex decision where it involves people issues more than technological issues. Identification of stakeholders is a key step during the process of ERP implementation, because if done improperly, it will lead to failure of the implementation project. The stakeholders are listed below:

● **Teachers**

Teachers are the key stakeholders of the college ERP. Because they are the one who manage, edit, update the contents of the database of students such as attendance, internal marks, CGPA etc...

It also helps them to assign their class to other teachers when they are on leave. This makes it easier to identify who among them are free to take the class at that

time. So this software helps them reduce their overhead and make their tasks easier and simple.

- **Administrator**

College administrator is responsible for maintaining the database of the college. They will have the privilege to modify the database i.e., to add/remove students/teachers/staff, update information regarding each of these. It is their responsibility to maintain the database of students who pass out from the college and who freshly get admission to the college. So the Administrator plays a major role in the ERP.

Viewpoints

- **Teachers viewpoint**

For a teacher, this software must be easy to use. It should be easy to find different modules like attendance, leave module, internals marks, result etc...Teachers are the one who update the contents of the database, so it should be updated to save and modify It.

- **Students viewpoint**

A student can only view the information about himself, other than that everything will be hidden from them. They will not have the option to edit anything. So the graphical user interface must be good.They expect it to be functional.

- **Administrator's viewpoint**

Administrators will have the privilege to view all the information about the college. They will have the option to track goals like, Average marks of all the students in a subject, Average attendance of all the students of a class etc...

Elicitation

When we started the project, we decided to collect the information from a couple of stakeholders like teachers, administrators, students and parents. They stated their role in the ERP system, their problems, likes and dislikes, problems they are facing with the software and how it is implemented

Can you explain the attendance entry process in detail?

Generally, just like the students, even teachers have their own user ID and attendance purposes in a hierarchical manner. First there will be two columns: class and subjects. Under the class column there will be a list of all the classes allotted to the faculty. On the other column there are subjects which are further divided into theoretical subjects which are of 4 credits and integrated subjects which are of 5 credits for the university batch students. Since there are autonomous batch students who are yet to complete their degree there are separate columns reserved for them since their pattern is different from the university syllabus. They will be having theoretical subjects of 4 credits each and they will also be having separate lab sessions of 1.5 credits each. Since the credits of autonomous subjects vary from those of the subjects of the university subjects there must be changes in terms of attendance and the credits allocated for each subject.

Then there will be a column for the type of class. In this there will be further two types. Regular classes and alternate classes. Regular classes password for the login purpose. There will be a column reserved for are those which the faculty

handles for the allocated class as specified in the time-table. Alternate classes are those which the faculty handles in the absence of another faculty. When the faculty is on leave, it must be informed in the ERP such that the message goes to the professors and at the same time another teacher who is free. If the faculty wants to take extra classes due to the incompleteness of the courses, then they should inform the students in the forum about the extra class and they can handle it. Generally, for the teachers there are basically 4 types of leave.

1. Earned leave
2. Restricted leave.
3. Casual leave.
4. Sick leave.

What are some problems that you face with the current ERP system?

The problem with the ERP software is if the faculty applies for leave and wants to allocate the class to any other faculty, then the request goes to all the faculties of all the departments. This should not happen because other department faculty cannot handle the class for any other department i.e. If the faculty of the Computer Science department applies for the leave and if the request is sent, it must be sent to the faculties of the Computer Science department only and not for any other department like Civil, Mechanical, E&E, and so on.

When the faculty is inserting the attendance into the system, there must be a separate space for the faculty to fill what topics they have covered in the class. It will be time consuming for the faculty to enter the topic every time. So, for this purpose the software must be designed in such a way that it inserts the topic automatically. Firstly, all the topics and the duration for the faculty in which the faculty must cover must be mentioned. And then the faculty must investigate it and cover the syllabus according to the plan. This can also keep a track of the lecturer what they are teaching. If the doubts are raised by the students, then that would lead to shortage of time to cover the syllabus. So, for this the faculty can

have the freedom to extend the duration to cover those topics by handling extra classes when the students are free. For taking the extra class, the faculty must block in the time table and it must be visible to all the faculties of that class so that there would be no collision in handling the extra class. Once if the faculty enters the attendance and if they press the lock option, then there won't be any option to change the attendance of the students. Lastly, the teachers would like it if they could enter the attendance in the class itself. This would minimize the paperwork and they could update the details at any place and at any time.

What do you expect from the module that lets you enter the marks of the students?

There will be another section to enter the CIE of all the students. The internals will be for 20 marks and when the faculty enters it into the ERP, it must automatically convert it into 10 marks. Generally, there will be 5 events. There will be 3 internals, followed by two events such as quiz, project. If the student scores below 50% of the allocated marks in the subject, then there must be a warning message sent to the student to score more marks in the upcoming internals.

At the end of all the events if the student could not mark the 50 mark, then there will be a make-up test conducted by the faculty so that the student would be having another chance to come up to the mark of 50%. These make-up test marks must be altered with the minimum marks of the CIE scored. And the final CIE marks should be displayed and be stated that the student is eligible or not eligible to take up the Semester End Examination. If the student is not able to take up the CIE due to personal reasons or if he is representing the college in any form of the activity, then it must be brought into the notice of the lecturer and the leave can be availed. If the student is ill, then the medical certificate must be attested, and a letter must be sent to the HOD to take up re-test. After the faculty enters the CIE there must be an option to save the CIE marks. When the CIE marks are saved then the students will not be able to see the marks in their

marks. They can view their CIE only when the marks are locked by the faculty. If the faculty locks the CIE, then there would not be any chance to change the CIE. The CIE must be locked after confirming the marks with the students only.

As a student, what are some problems you are facing with the current ERP system?

The ERP status was not updated regularly, and they could not track their attendance status as the app would crash. The GUI that is used in the interface is not up to the mark. It is difficult to keep the track of the attendance and the CIE. It would be easy if the attendance would be shown in a calendar like format so that it would be convenient and can also keep a track of the status of the attendance. There should also be forums where the teacher and the students are active. This will help the students in many ways such as studies, assignments, projects and so on. There should be interaction with the student-student and student-teacher so that the students can clear their doubts with any teacher as well as any student at any point of time. The forum will also help the students in conveying the information to all the students at a faster rate. For the students who were in the supplementary batch, they could not attend the first few weeks of class as they had exams. But, in the ERP they were marked as absent which made their attendance drastically low. When the students are into college activities such as LCC sessions, IEEE sessions, representing our college in sports or any other activities then students are marked absent. There must be another way to handle these problems so that there will be justice for the students for their hard work.

Administrator

- 1. What are your requirements from the ERP system as an admin?**

As an administrator, they deal with a large amount of data and functions. The system must be modular with a simple interface. The admin performs many functions on the database. These include searching for a record, add, update and delete a record. Thus, their interface needs to be quick and searching for records in the huge database must be optimized.

Elaboration

For the College ERP project, there are many classes of end users. These include the college staff, students and admin. As mentioned in the elicitation section, we talked with several stakeholders of different classes and collected their requirements. The requirements of the different classes were diverse. Some of them were in unison and some were in conflict. Thus, elaboration and later negotiation is required.

College staff

College staff are key stakeholders and use the ERP system the most. Thus, it is essential to cater to their needs first. Among the staff there are several different roles. For each role, The ERP system will have a different view based on the requirements of that group. Among the staff the requirements of the various groups are described below.

Teaching staff

Teaching staff make up most of the staff. A teacher expects the ERP system to be easy to use, reliable and reduce the workload. Each teacher belongs to a department and is assigned to a class of students with a course. So, the teacher should only be able to view and manipulate the data of the students that they are assigned to.

The teachers' involvement In the ERP System, is to enter the attendance, the internal marks, the semester end examination marks. They will also have other features which include availing leave and managing a lecture plan for each course.

For Attendance management, the teachers expect a compact and functional interface. An interface where teachers use minimal effort to manage the attendance status of the students. The features expected for the attendance are the ability to enter the attendance to the entire class at once, edit the attendance of each individual student. Also, in the event of leave, they should be given an option of assigning the class to another teacher, who takes a course for the same class.

In the event of entering internal marks and semester end examination marks, the teacher enters the marks for each individual student. This is initially a draft and can be edited. The students review the marks and verify. If there are any mistakes, the student notifies the teacher and the mistakes are corrected. After a certain amount of time, when all the marks are confirmed, the marks are 'locked'. i.e., After locking, the marks cannot be changed. When a teacher applies for a leave, there are many options for different categories of leave. The first category is casual leave, this is for general purposes. Restricted leave can be availed only on specific days given by the college. Then, Sick leave is for when the teacher is ill. Lastly, earned leave is an option given to each teacher for a period of 15 days.

Head of Department (HOD)

The head of department is a part of the teaching staff but has special privileges. They manage the operations of each department. The HOD could still conduct courses for students. So, they will have all the features given to a teacher. Also, as the HOD, they will have access to the records of every teacher, courses, students who belong to the same department.

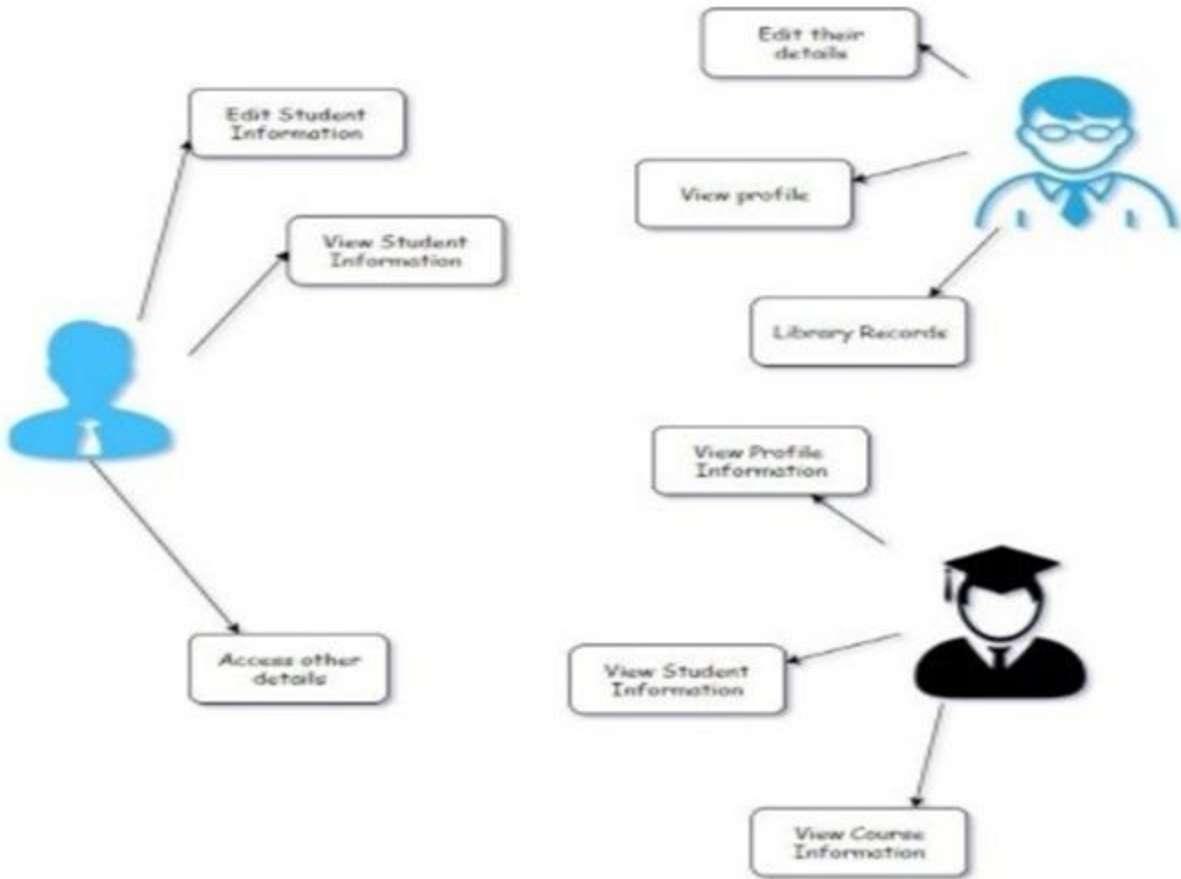
Technical staff

The technical staff provide technical support to the teaching staff. This includes technical support in laboratories such as maintaining the functioning of computers in labs, maintaining the equipment in their respective departments. Unlike the teaching staff, they do not conduct classes. Their role in the ERP system is to provide support to other staff through communication. Also, they have the feature to avail for leave just like the teachers.

Students

Students are another class of end users. In the ERP system, students can view information regarding their attendance status, internals marks, Semester end examination marks, notifications from the college administration etc. A student expects the ERP system to be aesthetic and functional. A student should only be able to view information about himself. Students generally want a nice graphical interface that provides a lot of information. In the case of attendance, they would like to see additional information other than just the number of classes attended. They would like to see day wise attendance so that they can keep a better record of their daily classes. Students also requested for a detailed view of attendance of each subject in a calendar form.

Lastly, another feature that students would like to see is the addition of communication and feedback capabilities in the ERP. Students find it tough to communicate with other students with common interests as there is no common communication medium in the college. A forum type infrastructure is required in the system.



Administrator

An admin holds all the privileges of the ERP system. The admin has access to all the databases in the system. These include student database, teacher database, courses database and several others. Their job is to maintain the systems and address the problems faced by the other users. The admin needs adequate resources and the right tools. The admin expects a simple interface where they can easily access the required information .

Negotiation

The College ERP system is vast with a lot of desired features and functionality. Each stakeholder gives their list of requirements. As a project with four group members, we do not have the resources and tools to implement all the

requirements. Thus, it is essential to find a balance among the various stakeholders where they can be satisfied with the outcome of the project. This is achieved through negotiation among the various classes of stakeholders.

It is in our interest to develop a Webapp that is functional, reliable, consistent and easy to use. We collected the requirements from the different stakeholders that include, the teaching staff, technical staff, students and the administration. We reviewed the list of requirements and made a list of feasible and infeasible requirements. We meet the stakeholders again and explain why some requirements were not feasible. For example, the leave module for the teachers can not be implemented as that feature has a lot of functionality that is beyond the scope of this project.

We also found that some requirements from different stakeholders were conflicting. For example, the students had requested for an option to appeal wrong marking of attendance or marks by the teacher. But, the teachers were against this feature as that would increase the burden on the teachers and there was also a possibility of false usage of this feature by the students. Considering both perspectives, we decided to agree with the teachers as this feature would displease teachers. The students were given the reason for not including their requirement and an agreement was reached.

The students wanted a social media type feature implemented on the ERP where the students from the college can communicate with each other and have a feed of the events in the college. While the feature would have been nice to see, it was beyond the scope of this project. We stated that such an advanced version of the requirements was not possible. But, an implementation of the feature on a smaller scope with lesser functionality was possible. Therefore, we negotiated the features until both parties were satisfied.

Specification

- **Purpose**

The purpose of this project is to develop a College Management System that helps the teachers and students in easier management of College activities such as attendance, marks.

Intended Audience and Reading Suggestions

This project is intended for staff and students of JSS Science and Technology University. This document has been made under the guidance of college professors. This document has been organized into Overall description followed by the features and then the functional and non-functional requirements. The document may be read to the desire of the reader.

Project Scope

The project is designed to help the teachers and students manage their college activities. It consists of relational databases of students, departments, faculty, courses of the entire university. Using these databases, various functions that include Attendance management, marks management and leave management are provided. Within attendance management, a teacher can enter the attendance status of each student for each course with their respective dates. Similar to attendance, Internal and Semester end marks can also be entered for each student.

Overall Description

- **Product Perspective**

This project is modeled based on the current ERP system in the college. Students and teachers face several problems while using the system. Therefore, we wanted to build a system that has a lesser number of features than the current system but, has more functionality.

- **Product Features**

- Each teacher will be able to enter attendance and marks for their respective students.
- Each student will be able to view the attendance status for their respective courses.
- The teachers will be able to apply for various types of leave directly through the system.
- The students will be able to Communicate and provide feedback to their teachers.
- The students will have access to a forum page where they can communicate with each other. The administrator will be able to view and update information such as departments, classes, teachers, students, and courses.

User Classes and Characteristics

There are several types of end users for the college ERP system. They are broadly divided as Students, Staff and the Administrator. Each of these classes have their own set of features.

1. The student should have the following features:

- View the Attendance status of the courses to which they are enrolled.
- View the Marks of the courses to which they are enrolled.
- View the notification from the college administrator.
- Communicate or give feedback to their respective teachers.
- Communicate with other students of the same university.

2. The staff should have the following features:

- Access to the information of all students that attend their courses.
- Add and edit the Attendance status of those students.
- Add and edit the exam marks of those students.
- Avail the different types of leave.

- Swap classes with other teachers who teach for the same class.

3. The administrator should have the following features:

- Add and update students, teachers and courses.
- Assign teachers and students to courses

Operating Environment

The operating environment for College ERP system are listed below:

- Operating System: Windows 10
- Database: MySQL database
- Front end: HTML/CSS/Bootstrap
- Back end: Django

System features

● Expected requirement: Student and staff information

Description and priority Information regarding students, teachers and courses are stored in the database. Every user can view only certain information based on their user class. For example, a teacher can view students and course information that they are handling. This feature is of high priority as the information must be viewed by only the authorized users.

Functional requirements

- Each user shall be able to view information in the database based on their user class.
- The administrator shall be able to view all the information in the database.

Normal requirement: Attendance and marks entry

Description and priority Attendance and marks entry is the main feature of the College ERP system. Hence, the priority is high. Teachers update the attendance and marks of the students who are part of her class. Students can view their respective Attendance and marks of the courses they have taken.

Teachers shall be able to view, update and edit the attendance and marks of the students, part of their class.

- Teachers shall be able to take extra classes, switch classes with other teachers.

Exciting requirement: Communication among students and teachers

Description and priority Students and teachers will be able to communicate with each other directly using the ERP system. Students may give their queries and feedback to a teacher and they may respond accordingly. The priority of this feature is low as cost of implementation could be very high. A simple version of this feature is to be implemented.

- Students shall be able to communicate with their teachers by sending personal messages.
- Students shall be able to communicate with other students through a forum section.

External Interface Requirements

• User Interfaces

The User interface is made using Bootstrap. Firstly, there will be a simple login page separate for students and teachers. Each student and teacher will have a unique interface. There will be a fixed sidebar with links to all the modules. The teachers will be able to view their respective students and update their attendance and marks using an effortless interface.

• Software Interfaces

The following is a list of software used in the making of the project.

Operating System: We have chosen Windows operating system for its best support and user- friendliness.

Django: We have chosen to use Django for the back-end of the website as Django is a simple python framework and is suitable for beginners.

- Database: We are using SQLite database, which comes as default with Django.

Communications Interfaces This project is to be deployed on an online website. All the users can connect to the database server from anywhere and have access to their information.

Non-functional requirements

- **Safety requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

- **Security requirements**

The database contains sensitive information of all the students and staff. Therefore, optimal security measures must be taken to ensure data is safe from unauthorized users.

- **Software Quality Attributes**

Availability: The users must always be able to view their information so that they can keep track Regularly.

Correctness: The information about attendance and marks must be correct to not feed wrong information to the users.

portability: The users access the ERP from various platforms such as desktops and mobile phones.

The webapp must be portable to all platforms and the user experience must be optimal.

Validation

Requirements validation examines the specification to ensure that all software requirements have been stated unambiguously, so that inconsistencies, omissions, and errors have been detected and corrected.

This checklist is a list of questions that helps us to validate our requirements. They are as follows: Are requirements stated clearly? Can they be misinterpreted?

A: There will be a chance of misinterpreting the requirements specified by the stakeholders. but we have collected requirements from many sources and those requirements are understood correctly. Is the source (e.g., a person, a regulation, a document) of the requirement identified? Has the final statement of the requirement been examined by or against the original source?

A: Yes all the sources of the requirements are correctly identified. And all the requirements are Verified. Does the requirement violate any system domain constraints?

A: Those requirements violating the system domain constraints were omitted during the negotiation of requirements. So no requirements are violating the system domain constraint. Is the requirement testable?

A: All the requirements collected are unambiguous, clear and precise. This makes the requirements testable. Is the requirement traceable to any system model that has been created?

A:Yes the requirement is traceable i.e.,the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)

Requirements Management

Requirements management can be defined as a process of eliciting, documenting, organizing, and controlling changes to the requirements. Generally, the process of requirements management begins as soon as the requirements document is available, but 'planning' for managing the changing requirements should start during the requirements elicitation process.

The essential activities performed in requirements management are listed below.

1. Recognizing the need for change in the requirements
2. Establishing a relationship amongst stakeholders and involving them in the requirements engineering process
3. Identifying and tracking requirements attributes.

Requirements management enables the development team to identify, control, and track requirements and changes that occur as the software development process progresses. Other advantages associated with the requirements management are listed below.

Better control of complex projects: This provides the development team with a clear understanding of what, when, and why the software is to be delivered. The resources are allocated according to user-driven priorities and relative implementation effort.

Improved software quality: This ensures that the software performs according to the requirements to enhance software quality. This can be achieved when the developers and testers have a precise understanding of what to develop and test.

Reduced project costs and delays: This minimizes errors early in the development cycle as it is expensive to 'fix' errors at the later stages of the development cycle. As a result, the project costs also reduce.

Improved team communication: This facilitates early involvement of users to ensure that their needs are achieved.

An efficient requirement change management process undergoes a number of stages for changes to the requirements. These stages are listed below.

1. Problem analysis and change specification: The entire process begins with identification of problems to the requirements. The problem or proposal is analyzed to verify whether the change is valid. The outcome of the analysis is provided to the 'change requester' and a more specific requirements change proposal is then made.
2. Change analysis and costing: The effect of a change requested on the requirement is assessed according to traceability information. The cost for this can be estimated on the basis of modification made to the design and implementation. After the analysis is over, a decision is made whether changes are to be made.
3. Change implementation: Finally, the changes are made to the requirements document, system design and implementation. The requirements document is organized in such a manner so that changes to it can be made without extensive rewriting. Minimizing the external references and making document sections modular achieves changeability in the document. By doing this, individual sections can be changed and replaced without affecting other parts of the document.

System Design

Various Design concepts and processes were applied to this project. Following concepts like separation of concerns, the software is divided into individual modules that are functionally independent and incorporate information hiding. The software is divided into 3 modules which are students, teachers and administrators. We shall look at each module in detail.

Student

Each student belongs to a class identified by semester and section. Each class belongs to a department and is assigned a set of courses. Therefore, these courses are common to all students of that class. The students are given a unique username and password to login. Each of them will have a different view. These views are described below.

- **Student information**

Each student can view only their own personal information. This includes their personal details like name, phone no, address etc. Also, they can view the courses they are enrolled in and the attendance marks of each of those.

- **Attendance information**

Attendance for each course will be displayed. This includes the number of attended classes and the attendance percentage. If the attendance percentage is below a specified threshold, say 75%, It will be marked in red otherwise it will be in green. There will also be a day wise attendance view for each course which shows the date and status. This will be presented in a calendar format.

- **Marks information**

There will be 5 events and 1 semester end examination for each course. The marks for each of these will be provided in the ERP system. Notifications and events This section is common to all students. Notifications are messages from the admin such as declaration of holidays, test time-table etc. The events and their details are specified here.

Teacher

Each teacher belongs to a department and is assigned to classes with a course. Teachers will also have a username and password to login. The different views for teachers are described below.

Information The teachers will have access to information regarding the courses and classes they are assigned to. Details of the courses include the credits, the syllabus plan. Details of the class include the department, semester, section and the list of students in each class. The teacher will also have access to information of students who belong to the same class as the teacher.

- **Attendance**

The teacher has the ability to add and also edit the attendance of each student. For entering the attendance, they will be given the list of students in each class and they can enter the attendance of the whole class on a day to day basis. There will be two radio buttons next to each student name, one for present and the other for absent. There will also be an option for extra classes. Teachers can edit the attendance of each student either for each student individually or for the whole class.

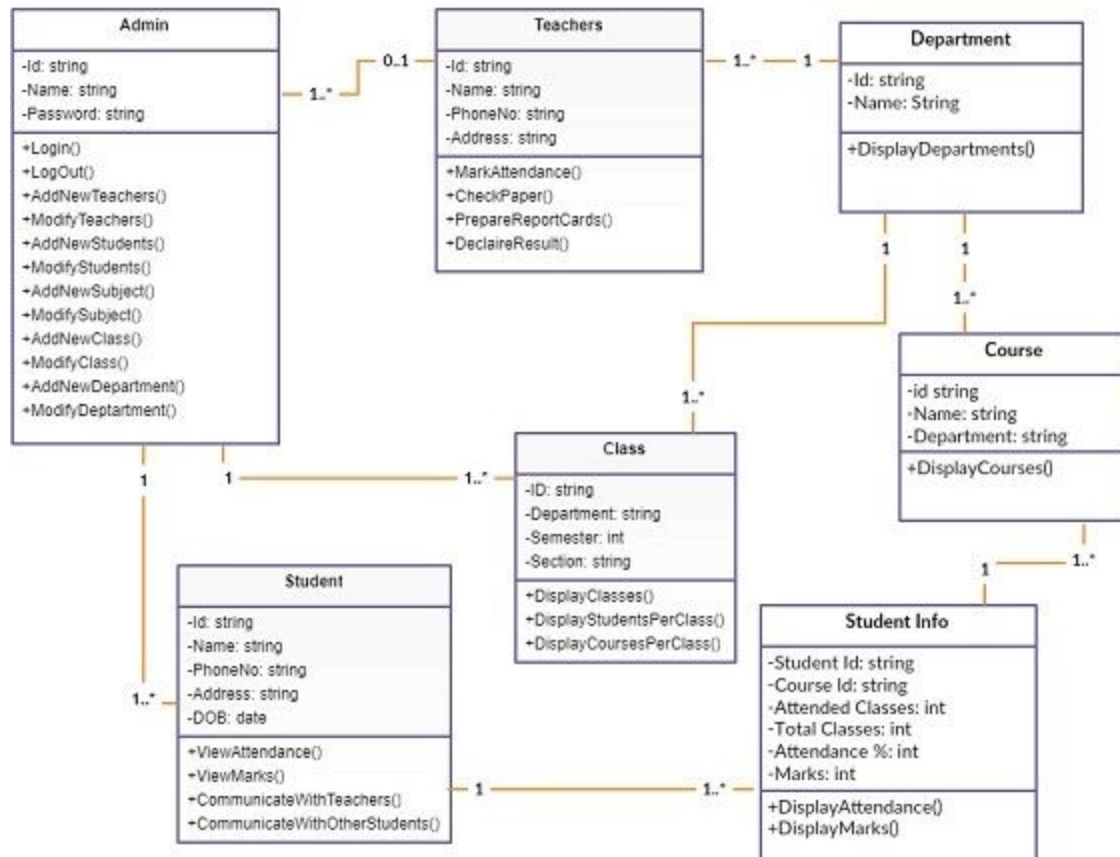
Marks The teacher can enter the marks for the 5 events and 1 SEE for each course they are assigned. They also have the ability to edit the marks in case of any changes. Reports such as the report cards including all the marks and CGPA of a student can be generated.

Administrator

The administrator will have access to all the information in the different tables in the database. They will access all the tables in a list form. They will be able to add an entry in any table and also edit them. The design of the view for the admin will provide a modular interface so that querying the tables will be optimized. They will be provided with search and filter features so that they can access data efficiently.

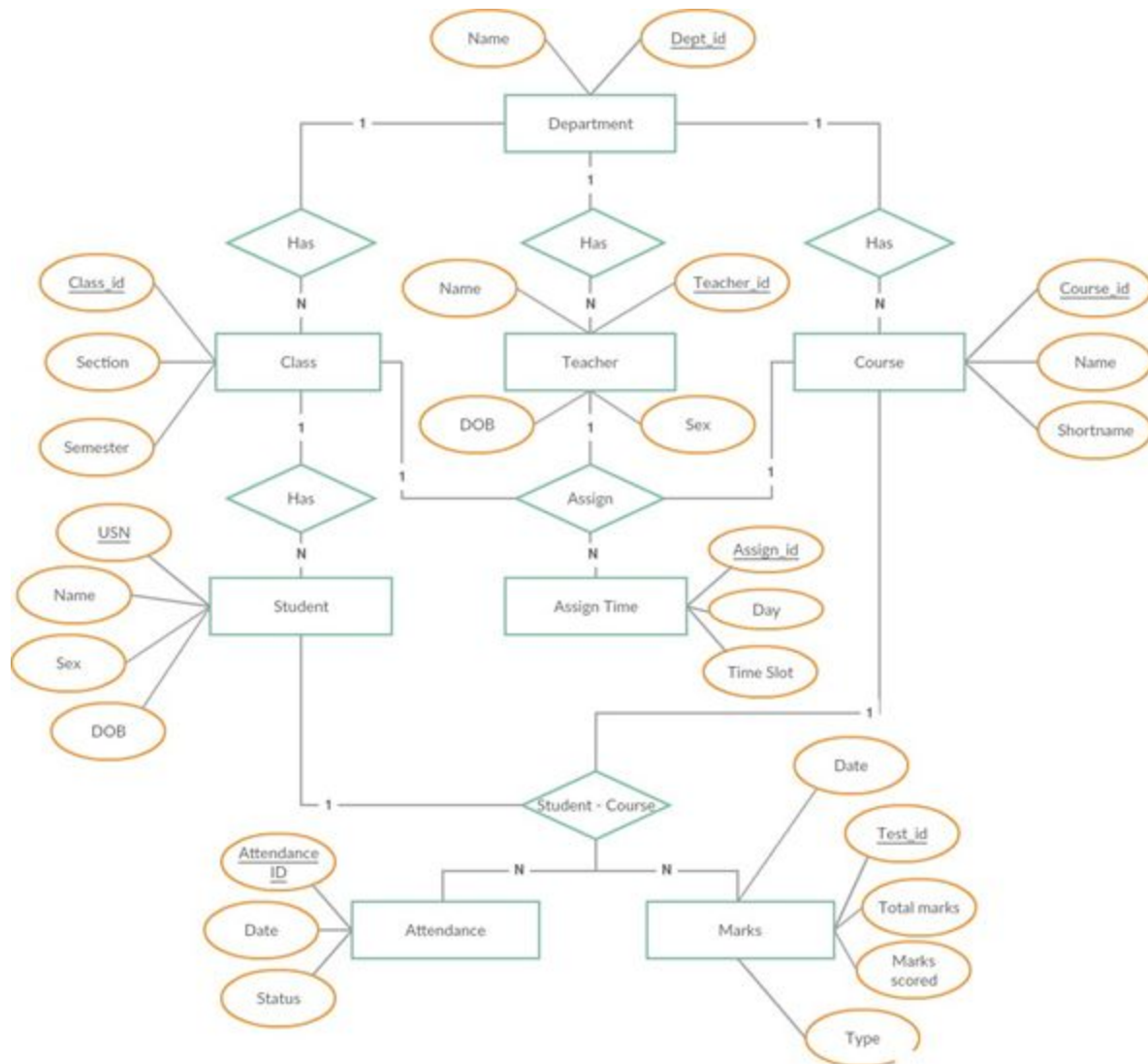
Class Diagram

The class diagram states the different classes involved in the software. For each class, a set of attributes and methods are included. The relationship between the classes are also specified. For example, the teacher class has the attributes Id, name, phone no, address and methods such as marking attendance, declaring marks and preparing report cards. Each instance of the teacher class belongs to a department. This is specified by the relationship between Teacher and Department classes.



Class diagram of college ERP

Entity Relationship Diagram



Entity Relationship diagram of college ERP

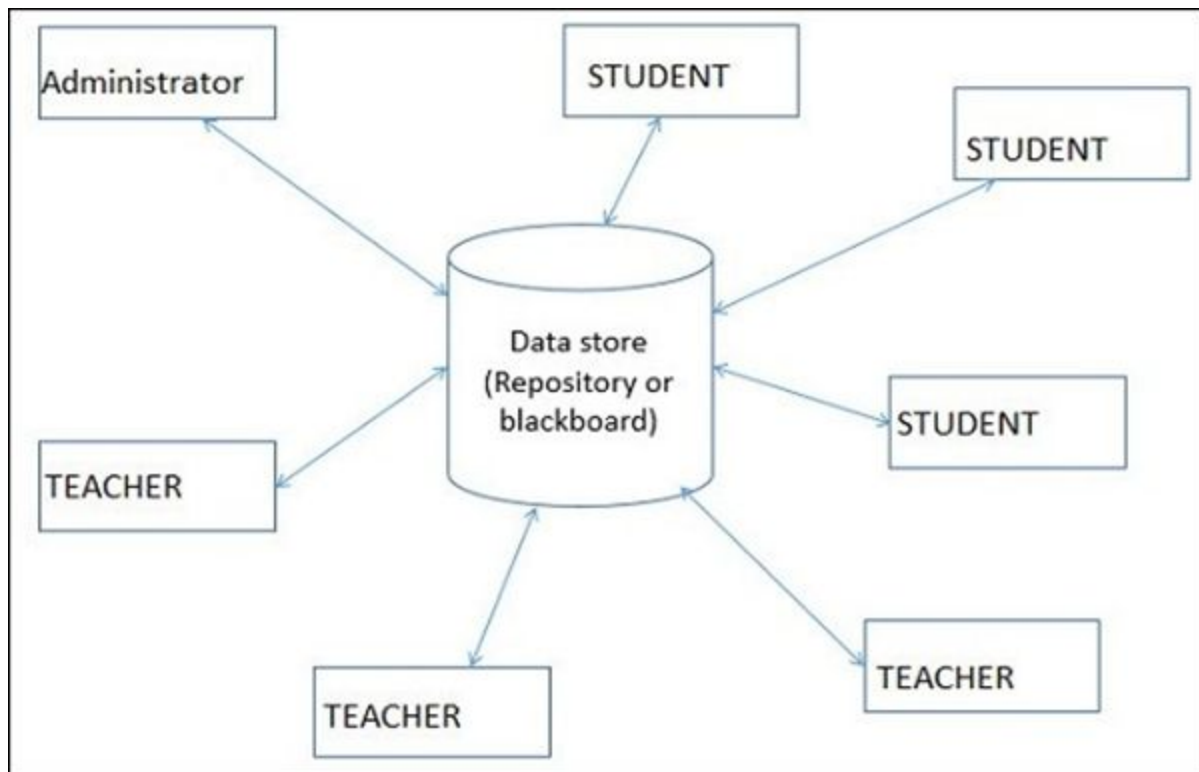
Architectural design

The ERP software requires the architectural design to represent the design of the software. Here we define a collection of hardware and software components and their interfaces to establish the framework for the development of this software. There exists a number of components of the system which are integrated to form a system. The set of connectors will help in coordination,

communication, and cooperation between the components. The ERP software is built for computer-based systems. It exhibits the data centric style of architecture.

- **Architectural style**

In the college ERP software, the database stores the data of all the students and faculties and the stored data is updated, added, deleted or modified. So it exhibits the data centric architectural style.



Data Centric architectural style

- **Central data**

Also known as data store or data repository, which is responsible for providing permanent data storage. It represents the current state. It stores the information of students, attendance of students and faculties of each day, salary of all the faculties etc...

- **Data accessors**

Data accessors are one of the components, they are also called as clients. A data accessor operates on the central data store, performs computations, and might put back the results. Which includes students, faculties and administrators. Students request to access the data from the repository and get the request serviced. Faculty members modify the data in the repository. Administrators can add or delete the clients.

Interface

Interface is the connecting component between a data repository and clients' client interacting with the data through the web server.

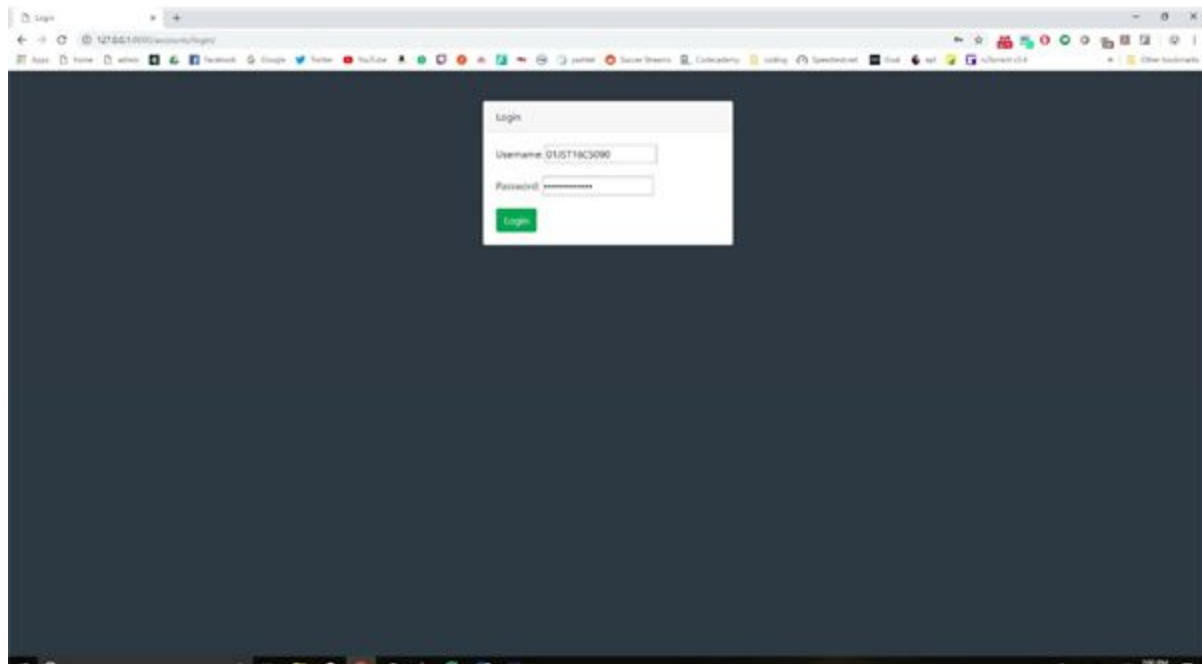
The operation of one client does not depend on the others. They are independent of each other. This data-centered architecture will promote integrability. This means that the existing components can be changed and new client components can be added to the architecture without the permission or concern of other clients. Addition or removal of students and faculties can be done without the permission of other students and faculties.

System Implementation

The college ERP system has three main user classes. These include the students, teachers and administrator. This section will explain in detail all the features and the working of those for each user class.

- **Student**
- **Login**

Each student in the college is assigned a unique username and password by the administrator. The user-name is the same as their USN and so is the password. They may change it later according to their wish.

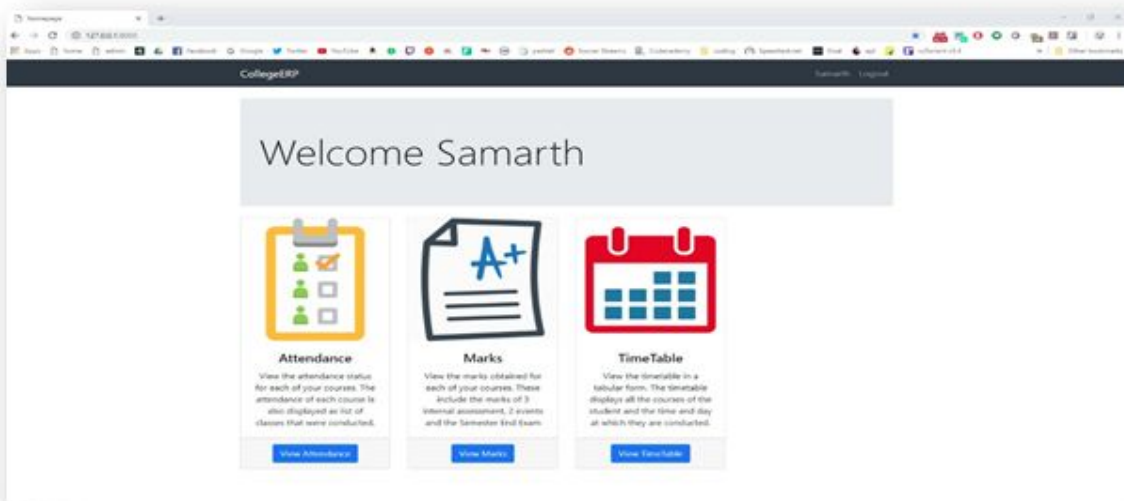


Student Login Page

Homepage

After successful login, the student is presented a homepage with their main sections, attendance, marks and timetable. In the attendance section the student can view their attendance status which includes the total classes, attended classes and the attendance percentage for each of their courses.

In the marks section, the student can view the marks for each of their courses out of 20 for 3 internal assessments, 2 events. Also, the semester end examination for 100 marks. Lastly, the times table provides the classes assigned to that student and day and time of each in a tabular form.



Student Home Page

Attendance

On the attendance page, there is a list of courses that is dependent on each student. For each course, the course id and name are displayed along with the attended classes, total classes and the attendance percentage for that particular course. If the attendance percentage is below 75 for any course, it is displayed in red denoting shortage of attendance, otherwise it is green. If there is any shortage, it specifies the number of classes to attend to make up for it. If you click on each course, it takes you to the attendance detail page.

Attendance Detail

This page displays more details for the attendance in each course. For each the course, there is a list of classes conducted and each is marked with the date, day and whether the student was present or absent on that particular date.

Course ID	Course name	Attended classes	Total classes	Attendance %	Classes to attend
CS510	Database Management System	4	6	66.67	2
CS520	UNIX	0	0	0	0
CS530	Software Engineering	8	10	80.0	0
CS540	Computer Networks	0	0	0	0
CS550	Language Processor	0	0	0	0
MA510	Linear Algebra	0	0	0	0

Student Attendance Detail Page

Marks

The Marks page is a table with an entry for each of their courses. The course id and name are specified along the marks obtained in each of the tests and exams. The tests include 3 internal assessments with marks obtained out of a total of 20, 2 events such as project, assignment, quiz etc., with marks out of 20. Lastly, one semester end exam with marks out of 100.

#	Date	Day	Status
1	Oct. 26, 2018	Friday	Present
2	Oct. 28, 2018	Monday	Absent
3	Oct. 31, 2018	Wednesday	Present
4	Nov. 1, 2018	Thursday	Present
5	Nov. 2, 2018	Friday	Present
6	Nov. 7, 2018	Wednesday	Present
7	Nov. 8, 2018	Thursday	Present
8	Nov. 9, 2018	Friday	Present
9	Nov. 12, 2018	Monday	Absent
10	Nov. 15, 2018	Thursday	Present

Student Marks Page

Timetable

This page is a table which lists the day and timings of each of the classes assigned to the student. The row headers are the days of the week and the column headers are the time slots. So, for each day, it specifies the classes in the time slots. The timetable is generated automatically from the assigned table, which is a table containing the information of all the teachers assigned to a class with a course and the timings of the classes.

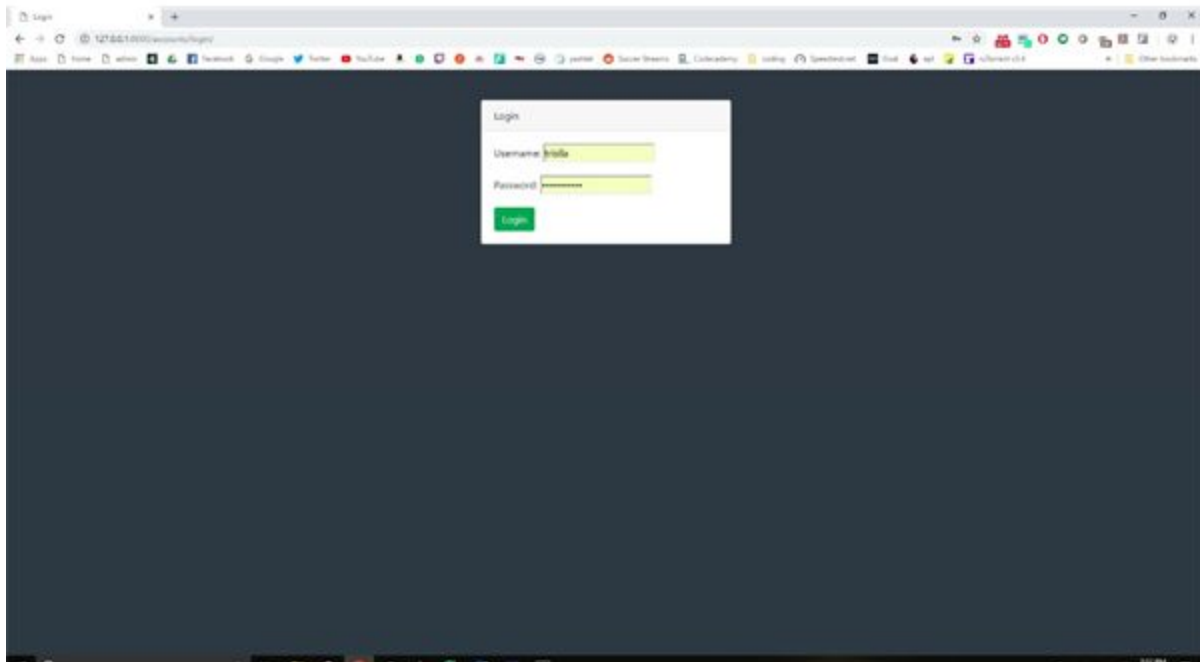
Course ID	Course name	Internal 1	Internal 2	Internal 3	Event 1	Event 2	SEE
CS310	Database Management System	0	0	0	0	0	0
CS320	UNIX	0	0	0	0	0	0
CS330	Software Engineering	0	15	10	15	15	0
CS340	Computer Networks	0	0	0	0	0	0
CS350	Language Processor	0	0	0	0	0	0
MA310	Linear Algebra	0	0	0	0	0	0

Student Timetable

Teacher

- **Login**

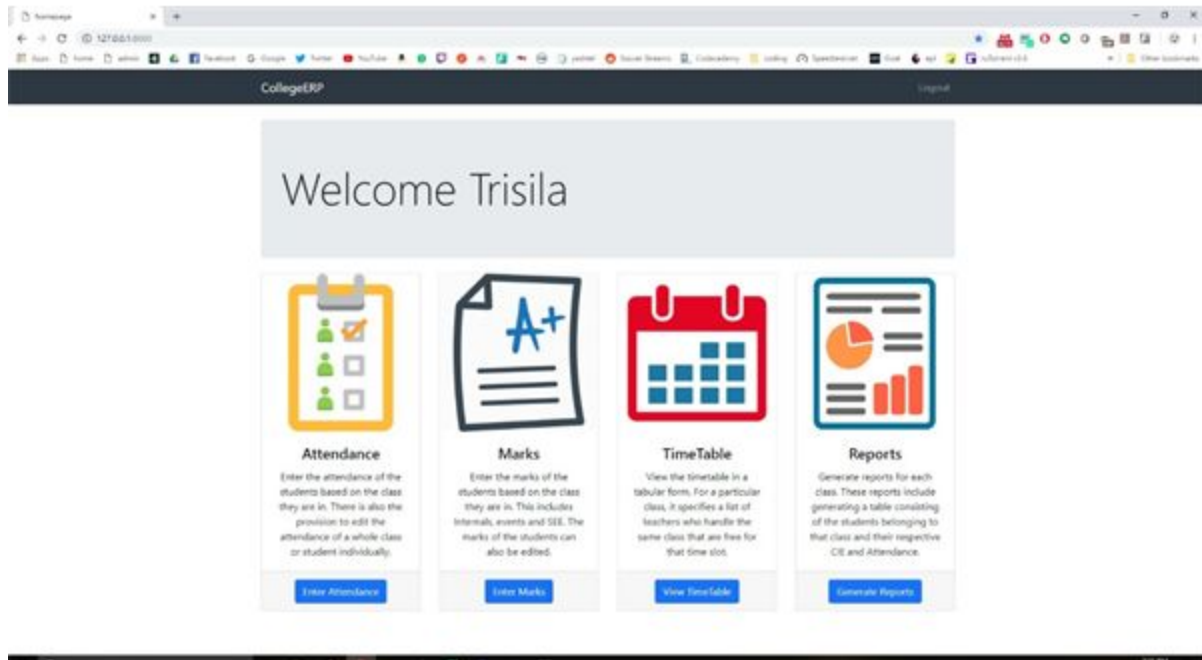
Each teacher in the college is assigned a unique username and password by the administrator. The Username is their teacher ID and the same for password. The teacher may change the password later.



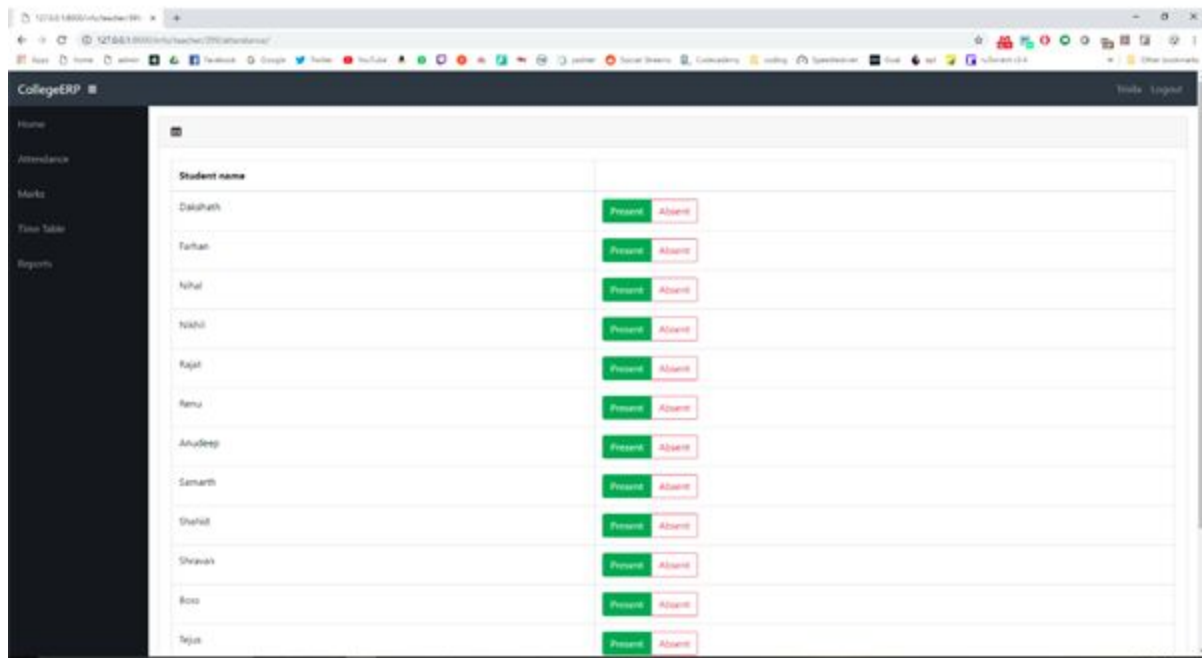
Teacher Login

Homepage

After successful login, the student is presented a homepage with their main sections, attendance, marks, timetable and reports. In the attendance section, the teacher can enter the attendance of their respective students for the days on which classes were conducted. There is a provision to enter extra classes and view/edit the attendance of each individual student. In the marks section, the teacher may enter the marks for 3 internals, 2 events and 1 SEE for each student. They can also edit each of the entered marks. The timetable provides the classes assigned to the teacher with the day and timings in a tabular form. Lastly, the teacher can generate reports for each of their assigned class.



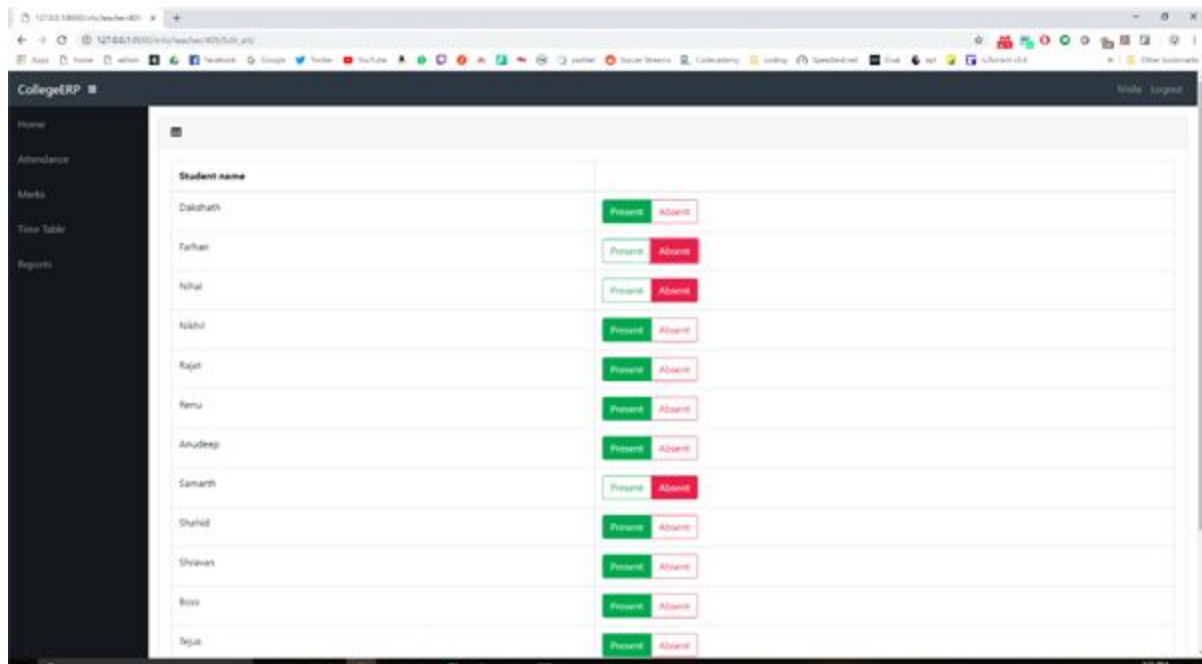
Teacher homepage



Entering attendance

Edit Attendance

After entering attendance, the teacher can also edit it. It is similar to a screen for entering attendance, only the entered attendance is saved and displayed. The teacher can change the appropriate attendance and save it.



Editing attendance

Student Attendance Details

The teacher can view the attendance detail of all their assigned students individually. That is, for all the conducted classes, it will display whether that student was present or absent. The teacher can also edit the attendance of each student individually by changing the attendance status for each conducted class.

The screenshot shows the 'Attendance' page in the CollegeERP system. The table lists 13 students (CS01 to CS13) with their respective attendance percentages. The 'Attendance %' column uses color coding: green for 90.0% and above, and red for 70.0% and below. The 'Classes to attend' column shows the number of classes a student needs to attend.

USN	Student name	Attended classes	Total classes	Attendance %	Classes to attend
CS01	Dakshini	8	10	80.0	0
CS02	Farhan	10	10	100.0	0
CS03	Nihal	7	10	70.0	3
CS04	Nikhil	8	10	80.0	0
CS05	Rajat	10	10	100.0	0
CS06	Renu	10	10	100.0	0
CS07	Anudhegi	8	10	80.0	0
CS08	Samarth	8	10	80.0	0
CS09	Shahid	10	10	100.0	0
CS10	Shravan	7	10	70.0	3
CS11	Ravi	10	10	100.0	0
CS12	Sejue	9	10	90.0	0
CS13	Vijeth	8	10	80.0	0

Attendance of students in a class

The screenshot shows the 'Software Engineering' page in the CollegeERP system. The table displays attendance details for an individual student, including the date, day, status (Present or Absent), and a 'Change' button for each entry.

#	Date	Day	Status	
1	Oct. 26, 2018	Friday	Absent	Change
2	Oct. 29, 2018	Monday	Present	Change
3	Oct. 31, 2018	Wednesday	Present	Change
4	Nov. 1, 2018	Thursday	Present	Change
5	Nov. 2, 2018	Friday	Present	Change
6	Nov. 7, 2018	Wednesday	Present	Change
7	Nov. 8, 2018	Thursday	Absent	Change
8	Nov. 9, 2018	Friday	Present	Change
9	Nov. 12, 2018	Monday	Present	Change
10	Nov. 15, 2018	Thursday	Present	Change

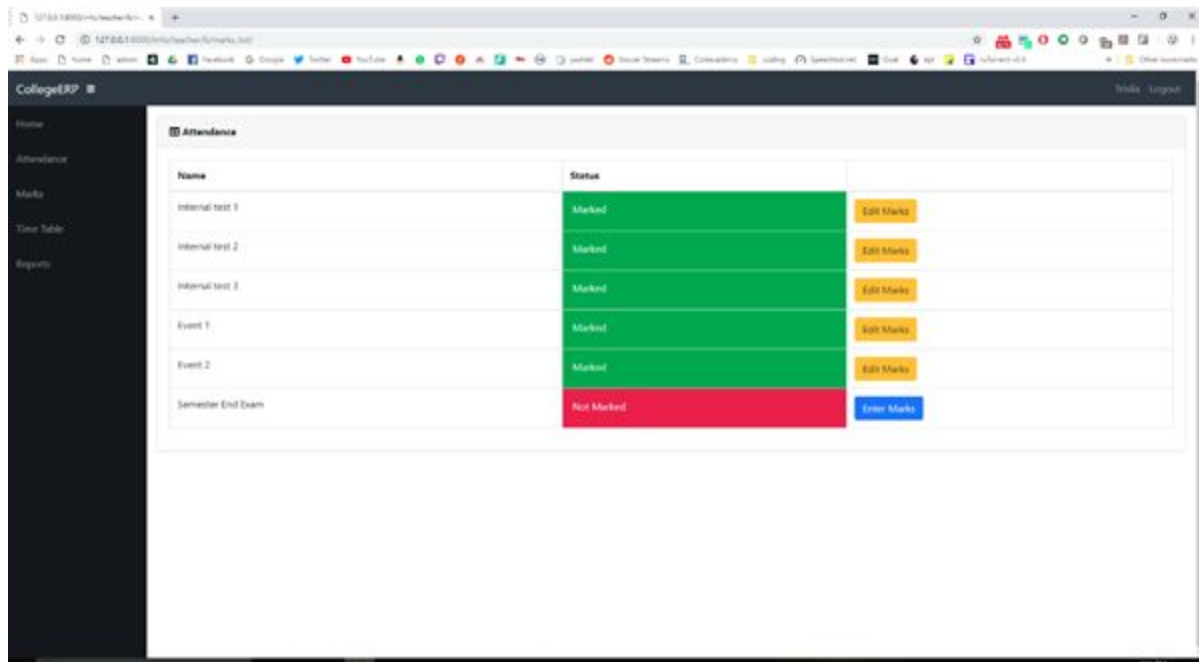
Attendance details of an individual student

Marks

On this page, the list of classes assigned to the teacher are displayed along with two actions for each class. These actions are,

- **Enter Marks**

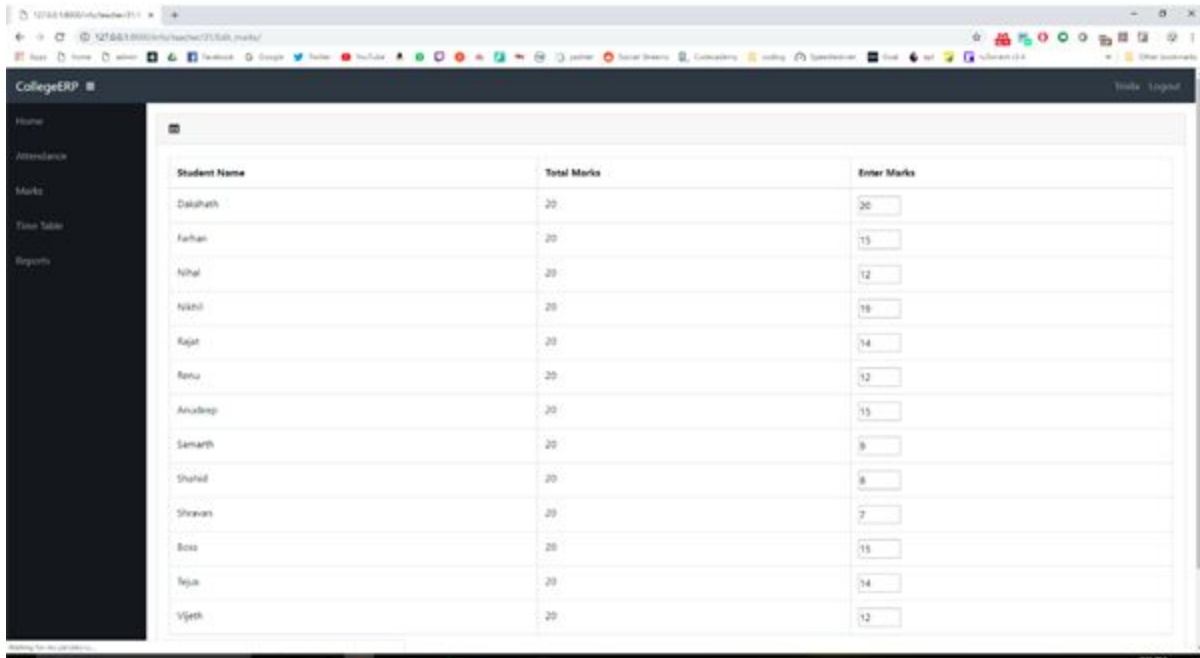
On this page, the teacher can enter the marks for 3 internal assessments, 2 events and one semester end exam. Initially all of them are marked red to denote that the marks have not been entered yet. Once the marks for a test is entered, it turns green. While entering the marks for a particular test, the list of students in that class is listed and marks can be entered for all of them and submitted. Once, the marks are submitted, the students can view their respective marks. In case there is a need to change the marks of any student, it is possible to edit the marks.



Entering marks

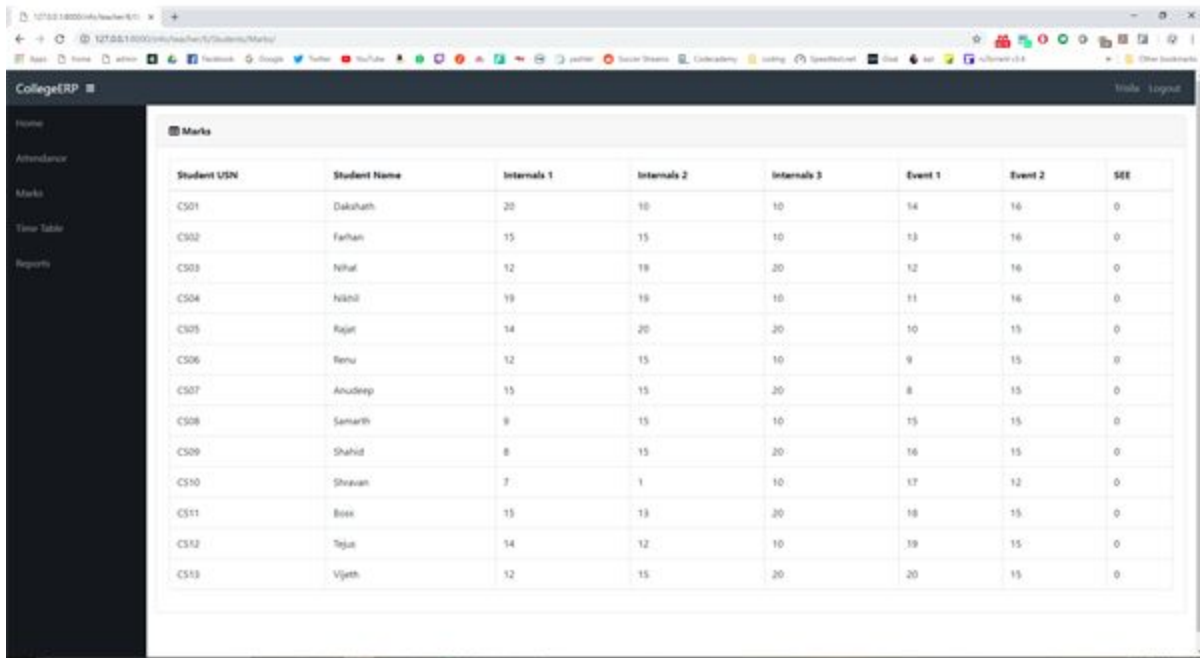
Edit Marks

Marks for a test can be edited. While editing, the list of students in that class is displayed along with already entered marks. The marks to be updated can be changed and submitted. The students can view this changed immediately.



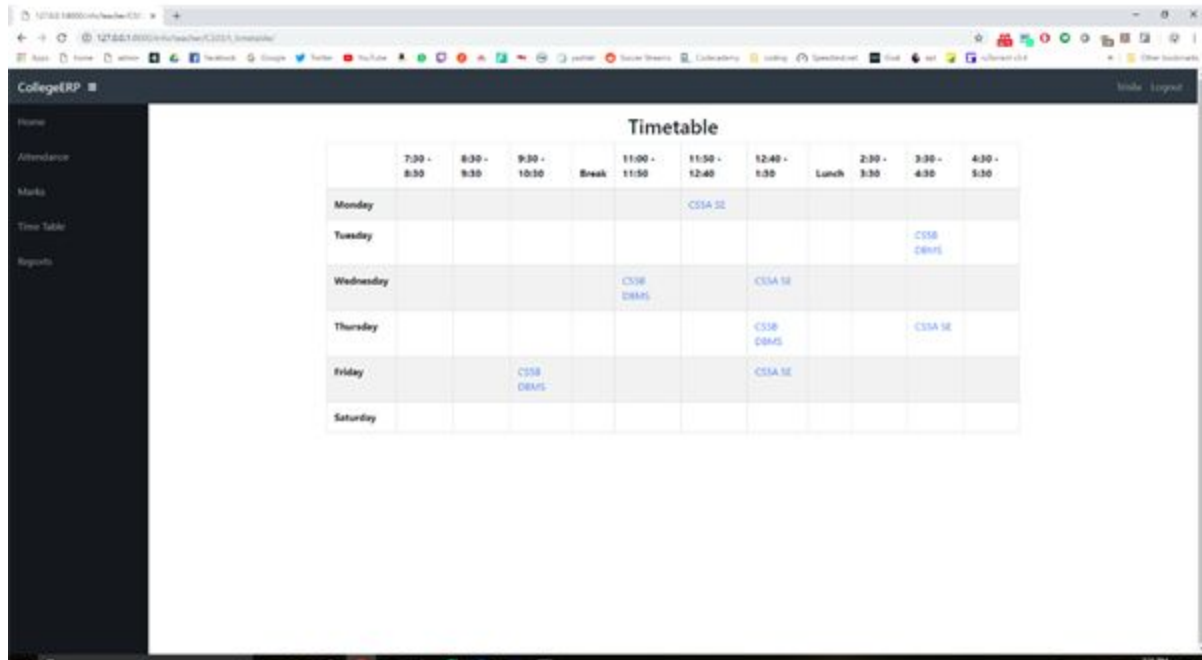
Student Name	Total Marks	Enter Marks
Dakshath	20	<input type="text" value="20"/>
Farhan	20	<input type="text" value="15"/>
Nihal	20	<input type="text" value="12"/>
Nikhil	20	<input type="text" value="19"/>
Rajat	20	<input type="text" value="14"/>
Renu	20	<input type="text" value="12"/>
Anudeep	20	<input type="text" value="15"/>
Samarth	20	<input type="text" value="9"/>
Shahid	20	<input type="text" value="8"/>
Shravan	20	<input type="text" value="7"/>
Boni	20	<input type="text" value="15"/>
Tejas	20	<input type="text" value="14"/>
Vijeth	20	<input type="text" value="12"/>

Editing marks



Student USN	Student Name	Internals 1	Internals 2	Internals 3	Event 1	Event 2	SEE
CS01	Dakshath	20	10	10	14	16	0
CS02	Farhan	15	15	10	13	16	0
CS03	Nihal	12	18	20	12	16	0
CS04	Nikhil	19	19	10	11	16	0
CS05	Rajat	14	20	20	10	15	0
CS06	Renu	12	15	10	9	15	0
CS07	Anudeep	15	15	20	8	15	0
CS08	Samarth	9	15	10	15	15	0
CS09	Shahid	8	15	20	16	15	0
CS10	Shravan	7	1	10	17	12	0
CS11	Boni	15	13	20	18	15	0
CS12	Tejas	14	12	10	19	15	0
CS13	Vijeth	12	15	20	20	15	0

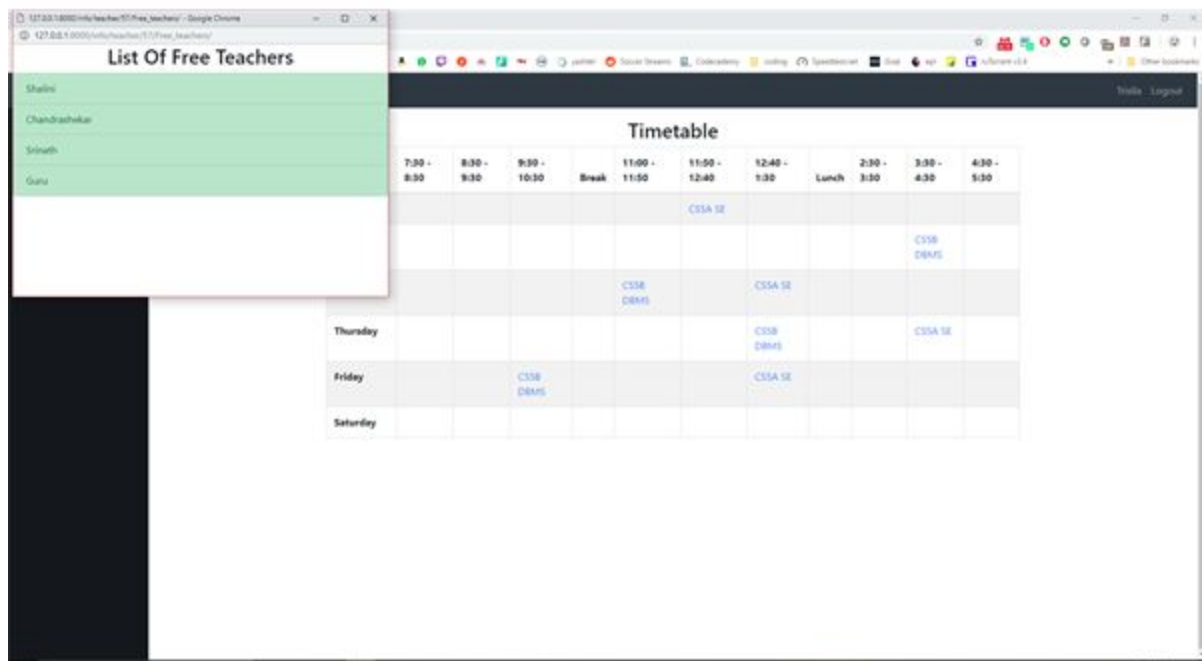
Marks of all the students in a class



The screenshot shows the 'CollegeERP' interface with a 'Timetable' view. The left sidebar contains links for Home, Attendance, Marks, Time Table, and Reports. The main area displays a weekly timetable grid. The columns represent time slots: 7:30-8:30, 8:30-9:30, 9:30-10:30, Break, 11:00-11:50, 11:50-12:40, 12:40-1:30, Lunch, 2:30-3:30, 3:30-4:30, and 4:30-5:30. The rows represent the days of the week from Monday to Saturday. The timetable shows the following assignments: Monday (CS5A SE at 11:50-12:40), Tuesday (CS5B DBMS at 3:30-4:30), Wednesday (CS5B DBMS at 11:00-11:50 and CS5A SE at 12:40-1:30), Thursday (CS5B DBMS at 12:40-1:30 and CS5A SE at 3:30-4:30), Friday (CS5B DBMS at 9:30-10:30 and CS5A SE at 12:40-1:30), and Saturday (empty).

	7:30 - 8:30	8:30 - 9:30	9:30 - 10:30	Break	11:00 - 11:50	11:50 - 12:40	12:40 - 1:30	Lunch	2:30 - 3:30	3:30 - 4:30	4:30 - 5:30
Monday						CS5A SE					
Tuesday										CS5B DBMS	
Wednesday					CS5B DBMS		CS5A SE				
Thursday							CS5B DBMS			CS5A SE	
Friday			CS5B DBMS				CS5A SE				
Saturday											

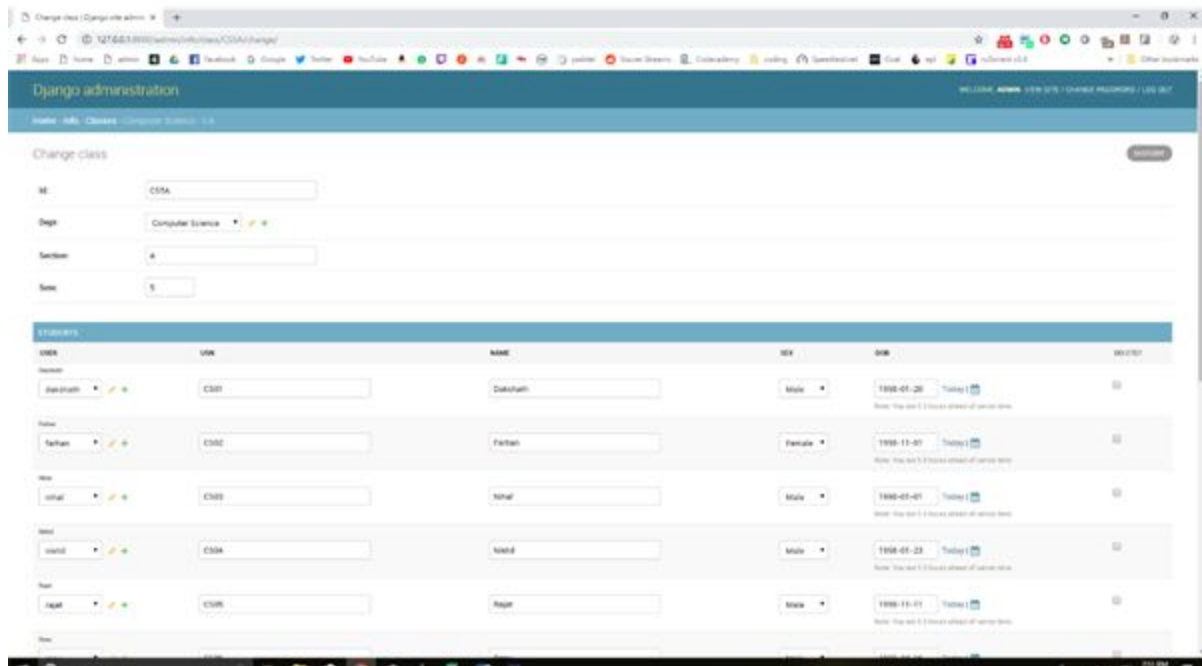
Teacher Timetable



The screenshot shows the 'CollegeERP' interface with a 'Timetable' view. A modal window titled 'List Of Free Teachers' is open on the left, displaying a list of teachers: Shalin, Chandrashekar, Sruthi, and Guna. The main area displays the same weekly timetable grid as the previous screenshot, showing the assignments for each day of the week.

	7:30 - 8:30	8:30 - 9:30	9:30 - 10:30	Break	11:00 - 11:50	11:50 - 12:40	12:40 - 1:30	Lunch	2:30 - 3:30	3:30 - 4:30	4:30 - 5:30
Monday						CS5A SE					
Tuesday										CS5B DBMS	
Wednesday					CS5B DBMS		CS5A SE				
Thursday							CS5B DBMS			CS5A SE	
Friday			CS5B DBMS				CS5A SE				
Saturday											

List of free teachers for a time slot



Admin students table page

System Testing and results analysis

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any breakdowns, therefore a package can be rejected even at this stage.

Testing methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

- **White Box Testing**

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these). White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

This project is implemented using python with the Django framework. The code consists of models and views which can be tested. Models define the tables stored in SQL and the relationship between the different tables using foreign keys. A view function, or “view” for short, is simply a Python function that takes a web request and returns a web response. This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image, etc.

Python also provides a file called test.py where we can write unit tests for the models and views. This is very useful as it automates the testing and we no longer have to manually test every page after there were any changes. The python code is pasted below and each test is explained using comments in the code.

```
def create_user(self,username='testuser',password='project123'):
self.client =Client()
return User.objects.create(username=username,password=password)
# testtocheckwhetheranobjectintheusertableiscreatedwithoutrrors
```

```

def test_user_creation(self):
    us =self.create_user()
    ut =self.create_user(username='teacher')
    s =Student(user=us,USN='CS01',name='test')
    s.save()
    t =Teacher(user=ut,id='CS01',name='test')
    t.save()
    self.assertTrue(isinstance(us, User))
    self.assertEqual(us.is_student, hasattr(us,'student'))
    self.assertEqual(ut.is_teacher, hasattr(ut,'teacher'))

# functionusedtocreatetestusers
def create_dept(self,id='CS',name='CS'):
    return Dept.objects.create(id=id,name=name)

# testtocheckwhetheranobjectintheusertableiscreatedwithouterrors
def test_dept_creation(self):
    d =self.create_dept()
    self.assertTrue(isinstance(d, Dept))
    self.assertEqual(d.__str__(), d.name)

# functionusedtocreatetestclass
def create_class(self,id='CS5A',sem=5,section='A'):
    dept =self.create_dept()
    return Class.objects.create(id=id,dept=dept,sem=sem,section=section)

# testtocheckwhetheranobjectintheclasstableiscreatedwithouterrors
def test_class_creation(self):
    c =self.create_class()
    self.assertTrue(isinstance(c, Class))
    self.assertEqual(c.__str__(), "%s:%d%s"%(c.dept.name,c.sem,c.section))

# functionusedtocreatetestcourse
def create_course(self,id='CS510',name='DataStruct',shortname='DS'):
    dept =self.create_dept(id='CS2')

```

```
return  
Course.objects.create(id=id,dept=dept,name=name,shortname=shortname)
```

```
# testtocheckwhetheranobjectinthecoursetableiscreatedwithouterrors
```

```
def test_course_creation(self):  
    c =self.create_course()  
    self.assertTrue(isinstance(c, Course))  
    self.assertEqual(c.__str__(), c.name)
```

```
# functionusedtocreateteststudent
```

```
def create_student(self,usn='CS01',name='samarth'):  
    cl =self.create_class()  
    u =self.create_user()  
    return Student.objects.create(user=u,class_id=cl,USN=usn,name=name)
```

```
# testtocheckwhetheranobjectinthestudenttableiscreatedwithouterrors
```

```
def test_student_creation(self):  
    s =self.create_student()  
    self.assertTrue(isinstance(s, Student))  
    self.assertEqual(s.__str__(), s.name)
```

```
# functionusedtocreatetestteacher
```

```
def create_teacher(self,id='CS01',name='teacher'):  
    dept =self.create_dept(id='CS3')  
    return Teacher.objects.create(id=id,name=name,dept=dept)
```

```
# testtocheckwhetheranobjectintheteachertableiscreatedwithouterrors
```

```
def test_teacher_creation(self):  
    s =self.create_teacher()  
    self.assertTrue(isinstance(s, Teacher))  
    self.assertEqual(s.__str__(), s.name)
```

```
# functionusedtocreatetestassign
```

```
def create_assign(self):  
    cl =self.create_class()  
    cr =self.create_course()
```

```

t =self.create_teacher()
return Assign.objects.create(class_id=cl,course=cr,teacher=t)
# testtocheckwhetheranobjectintheassigntableiscreatedwithouterrors
def test_assign_creation(self):
a =self.create_assign()
self.assertTrue(isinstance(a, Assign))

# views

# setupatestusersothatloginispossible
def setUp(self):
self.client =Client()
self.user =User.objects.create_user('test_user', 'test@test.com', 'test_password')

# testtoensureadmindoesn'thaveaccesstostudenttotteacherpage
def test_index_admin(self):
self.client.login(username='test_user', password='test_password')
response =self.client.get(reverse('index'))
self.assertContains(response, "youhavebeenloggedout")
self.assertEqual(response.status_code, 200)

# testtoensurestudentcanaccessonlythestudentpage
def test_index_student(self):
self.client.login(username='test_user', password='test_password')
s =Student.objects.create(user=User.objects.first(),USN='test',name='test_name')
response =self.client.get(reverse('index'))
self.assertContains(response, s.name)
self.assertEqual(response.status_code, 200)
test toensureteachercanaccessonlytheteacherpage
def test_index_teacher(self):
self.client.login(username='test_user', password='test_password')
s =Teacher.objects.create(user=User.objects.first(),id='test',name='test_name')
response =self.client.get(reverse('index'))
self.assertContains(response, s.name)
self.assertEqual(response.status_code, 200)

```



```
# testforresponse"studenthasnocourses"inthewebsite
```

```
# whenstudenthasn'tbeenassignedanycourse
```

```
def test_no_attendance(self):  
    s =self.create_student()  
    self.client.login(username='test_user', password='test_password')  
    response =self.client.get(reverse('attendance',args=(s.USN,)))  
    self.assertContains(response, "studenthasnocourses")  
    self.assertEqual(response.status_code, 200)
```

```
# testwhichassignsstudentacourseandtestswheethertheattendanceforthe
```

```
# sameisdisplayedonthewebsite
```

```
def test_attendance_view(self):  
    s =self.create_student()  
    self.client.login(username='test_user', password='test_password')  
    Assign.objects.create(class_id=s.class_id, course=self.create_course(),  
    teacher=self.create_teacher())  
    response =self.client.get(reverse('attendance',args=(s.USN,)))  
    self.assertEqual(response.status_code, 200)  
    self.assertQuerysetEqual(response.context['att_list'],  
    ['<AttendanceTotal: AttendanceTotalobject(1)>'])
```

```
# testforresponse"studenthasnoattendance"ontheattendancedetailpage
```

```
# whenteacherhasn'tmarkedanyattendanceforthatcourseyet.
```

```
def test_no_attendance__detail(self):  
    s =self.create_student()  
    cr =self.create_course()  
    self.client.login(username='test_user', password='test_password')  
    resp =self.client.get(reverse('attendance_detail',args=(s.USN,cr.id)))  
    self.assertEqual(resp.status_code, 200)
```

```
self.assertContains(resp, "studenthasnoattendance")
```

```
# testwhichmarksonattendanceforthestudentandcourseandtestswheether
```

```
# itisdisplayedproperlyintheattendancedetailpage.
```

```
def test_attendance__detail(self):  
    s =self.create_student()
```

```

cr =self.create_course()
Attendance.objects.create(student=s, course=cr)
self.client.login(username='test_user', password='test_password')
resp =self.client.get(reverse('attendance_detail',args=(s.USN,cr.id)))
self.assertEqual(resp.status_code, 200)
self.assertQuerysetEqual(resp.context['att_list'],
['<Attendance: '+s.name+'.'+cr.shortname+'>'])

```

Result ofTesting

```

py11 D:\Projects\django\CollegeM...python manage.py test -v2
creating test database for alias 'default' ('file:memory://default?mode=memory&cache=shared')...
Operations to perform:
Synchronize unmigrated apps: messages, staticfiles
Apply all migrations: admin, auth, contenttypes, info, sessions
Synchronizing apps without migrations:
Creating tables...
Running deferred SQL...
Running migrations:
Applying contenttypes.0001_initial... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0001_initial... OK
Applying auth.0002_alter_username_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_and_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying info.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_index_ordering... OK
Applying info.0001_auto_20181009_2000... OK
Applying info.0002_auto_20181009_2011... OK
Applying info.0003_auto_20181009_2014... OK
Applying info.0004_teacher_user... OK
Applying info.0005_auto_20181009_2210... OK
Applying info.0006_auto_20181011_1312... OK
Applying info.0007_auto_20181011_1314... OK
Applying info.0008_auto_20181011_1617... OK
Applying info.0009_auto_20181011_1618... OK
Applying info.0010_auto_20181017_2000... OK
Applying sessions.0001_initial... OK
System check identified no issues (0 silenced)
test_assign_creation (info.tests.InfoTest) ... OK
test_attendance_detail (info.tests.InfoTest) ... OK
test_attendance_line (info.tests.InfoTest) ... OK
test_class_creation (info.tests.InfoTest) ... OK
test_course_creation (info.tests.InfoTest) ... OK
test_dept_creation (info.tests.InfoTest) ... OK
test_index_admin (info.tests.InfoTest) ... OK
test_index_themes (info.tests.InfoTest) ... OK
test_index_teacher (info.tests.InfoTest) ... OK
test_no_attendance (info.tests.InfoTest) ... OK
test_no_attendance_detail (info.tests.InfoTest) ... OK
test_student_creation (info.tests.InfoTest) ... OK
test_teacher_creation (info.tests.InfoTest) ... OK
test_user_creation (info.tests.InfoTest) ... OK

Ran 14 tests in 3.000s

Destroying test database for alias 'default' ('file:memory://default?mode=memory&cache=shared')...
py11 D:\Projects\django\CollegeM...

```

References

1. Elmasri and Navathe: Fundamentals of Database Systems, 7th Edition, Pearson Education, 2016.
2. Ian Sommerville: Software Engineering, 10th edition, Pearson Education Ltd, 2015.
3. Roger S Pressman: Software Engineering- A Practitioner's approach, 8th edition, McGraw-Hill

Publication, 2015.

4. <https://en.wikipedia.org/wiki/Requirements-engineering>
5. <https://web.cs.dal.ca/~hawkey/3130/srs-template-ieee.doc>
6. <http://www.ntu.edu.sg/home/cfcavallaro/Reports/Report%20writing.htm>Top
7. [https://en.wikipedia.org/wiki/Class diagram](https://en.wikipedia.org/wiki/Class_diagram)
8. <https://www.djangoproject.com/>
9. <https://getbootstrap.com/>
10. <https://www.tutorialspoint.com/>
11. <https://creatly.com/>
12. <https://www.overleaf.com/project>



Thank You