

Summer Training Report

On

College ERP System

Project Report Submitted in Partial Fulfilment of
the Requirements for the Award of
Bachelor in Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By
Vivsvaan Sharma
(Roll No: CO16362)

Under the portal
(<https://www.udemy.com/>)

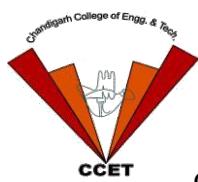


CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY
(DEGREE WING)

Government Institute under Chandigarh (UT) Administration, Affiliated to Punjab University,
Chandigarh

Sector-26 Chandigarh PIN-160019

June, 2019



CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)
Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh
Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943
Website: www.ccet.ac.in | Email: principal@ccet.ac.in | Fax. No. :0172-2750872

Department of Computer Sc. & Engineering

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled “College ERP System”, in fulfilment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering and Technology (Degree Wing) affiliated to Punjab University, Chandigarh, is an authentic record of my own work carried out during my degree under the Udemy Portal. The work reported in this has not been submitted by me for award of any other degree or diploma.

Date: 13-12-2019

Vivsvaan Sharma
CO16362

Place: Chandigarh

Certificate I

Certificate of Completion

*This is to certify that Vivsvaan Sharma
successfully completed 18.5 hours of Django Core
| A Reference Guide to Core Django Concepts
online course on June 25, 2019*

Justin Mitchel

Justin Mitchel, Instructor

&



Certificate No.: UC-1KTQ57PZ
Certificate url: [udemy.com/UC-1KTQ57PZ](https://www.udemy.com/UC-1KTQ57PZ)



Certificate II

Certificate of Completion

*This is to certify that Vivasvan Sharma
successfully completed 21 hours of The Ultimate
MySQL Bootcamp: Go from SQL Beginner to
Expert online course on July 1, 2019*

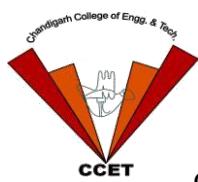
Colt Steele Ian Schoonover
Colt Steele, Instructor Ian Schoonover, Instructor

&

Udemy



Certificate no. UC-P2UDQFPYW
Certificate url: ude.my/UC-F2UDQFPYW



CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)
Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh
Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943
Website: www.ccet.ac.in | Email: principal@ccet.ac.in | Fax. No. :0172-2750872

Department of Computer Sc. & Engineering

ACKNOWLEDGEMENT

It is a great pleasure to present this Summer Training Report on College ERP System. I have taken efforts in this Project. However, it would not have been possible without the kind support and help of Mentors and fellow mates on Udemy Portal.

I am highly indebted to Chandigarh College of Engineering & Technology (Degree Wing) for their guidance and constant supervision as well as for providing necessary information regarding the practical & also for their support in completing this report.

Mentors on udemy portal taught me all the basic concepts required for the project and guided me through each step of building the programs whenever I was stuck.

I would like to express my special gratitude and thanks to institution (C.C.E.T.) persons for giving me such attention and time.

I would also like to thank the University for including this Summer Training as a part of my curriculum.

Vivsvaan Sharma
CO16362
CSE, 7th Semester



Department of Computer Sc. & Engineering

ABSTRACT

Enterprise Resource Planning system, popularly known as ERP system has become very popular as an enterprise management software tool. By using Existing System for accessing information from files is a difficult task and there is no quick and easy way to keep the records of students and staff. Lack of automation is also there in the Existing System. The aim of Our System is to reduce the workload and to save significant staff time. Title of the project as College ERP System is the system that deals with the issues related to a particular institution. It is the very useful to the student as well as the faculties to easy access to finding the details. The college ERP provides appropriate information to users based on their profiles and role in the system. This project is designed keeping in view the day to day problems faced by a college system. The fundamental problem in maintaining and managing the work by the administrator is hence overcome. This System provide the automate admissions no manual processing is required. This is a paperless work. It can be monitored and controlled remotely. It reduces the man power required. It provides accurate information always. All years together gathered information can be saved and can be accessed at any time. The data which is stored in the repository helps in taking intelligent decisions by the management providing the accurate results. The storage facility will ease the job of the operator. Thus the system developed will be helpful to the administrator by easing his/her task providing the accurate results. The storage facility will ease the job of the operator. This project is successfully implemented with all the features and modules of the college management system as per requirements.

TABLE OF CONTENTS

Students Declaration	i
Certificate by the guide	ii
Acknowledgement	iv
Abstract	v
Table of content	vi
List of tables	viii
List of figures	ix
Chapter 1 – Introduction to Python	1
1.1 Installation of Python	1
1.2 Salient Features of Python	2
1.3 Basic Syntax of Python	3
Chapter 2 – Using Variables in Python	6
2.1 Data Types and Variables	6
2.2 Standard Data Types	6
2.3 Data Type Conversions	9
Chapter 3 – Basics of Programming in Python	10
3.1 Basic Operators	10
3.2 Decision Making	10
3.3 Loops in Python	11
3.4 Functions in Python	13
3.5 Function Arguments	13
3.6 Scope of Variable	15
Chapter 4 – OOPS in Python	16
4.1 OOPS Concept	16
4.2 Python Class and Objects	18
4.3 Python Constructor	18
Chapter 5 – Django Framework	19
5.1 Introduction	19
5.2 Installing Django	19
5.3 Getting Started with Django	20

Chapter 6 – Project – College ERP System	25
6.1 Introduction	25
6.2 Overall Description	26
6.3 System Design	27
6.4 Modules in the system	32
6.5 Screenshots of the Implemented System	36
6.6 Conclusion	43
References	44

LIST OF TABLES

FIGURES	PAGE
Table 1.1 Reserved Words	4
Table 2.1 Data Type Conversion Fuctions	9

LIST OF FIGURES

FIGURES	PAGE
Fig 3.1 If Condition	11
Fig 3.2 Loops	12
Fig 6.1 Use Case Diagram of College ERP System	29
Fig 6.2 Class Diagram of College ERP System	30
Fig 6.3 ER Diagram of College ERP System	31
Fig 6.4 Login Page	36
Fig 6.5 Student Home Page	36
Fig 6.6 Student Attendance Page	37
Fig 6.7 Student Attendance Detail Page	37
Fig 6.8 Student Marks Page	38
Fig 6.9 Student Time Table Page	38
Fig 6.10 Teacher Home Page	39
Fig 6.11 Entering Attendance	39
Fig 6.12 Editing Attendance	40
Fig 6.13 Attendance of Student in Class	40
Fig 6.14 Attendance Details of an individual Student	41
Fig 6.15 Editing Marks	41
Fig 6.16 Marks of all the Students	42
Fig 6.17 CIE and Attendance of the class	42

Chapter 1: Introduction to python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1.1 Installation of Python

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing –

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you need to install.
- To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

Programs and other executable files can be in many directories; so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The path variable is named as PATH in Unix or Path in Windows

1.2 Salient Features of Python

Python's features include –

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read – Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain – Python's source code is fairly easy-to-maintain.
- A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.
- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

1.3 Basic Syntax of Python

1.3.1 Python identifiers

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, \$, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.

Here are naming conventions for Python identifiers –

- Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
- Starting an identifier with a single leading underscore indicates that the identifier is private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

1.3.2 Reserved Words

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Table 1.1 – Reserved Words

1.3.3 Lines and Indentation

Python provides no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.

The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. For example –

```
if True:  
    print "True"  
else:  
    print "False"
```

However, the following block generates an error –

```
if True:  
print "Answer"  
print "True"  
else:  
print "Answer"  
print "False"
```

Thus, in Python all the continuous lines indented with same number of spaces would form a block.

1.3.4 Multi-Line Statements

Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue. For example

```
total = item_one + \  
       item_two + \  
       item_three
```

Statements contained within the [], {}, or () brackets do not need to use the line continuation character. For example –

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

1.3.6 Comments in Python

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores them.

```
# First comment
print "Hello, Python!" # second comment
```

You can comment multiple lines as follows –

```
# This is a comment.
# This is a comment, too.
# This is a comment, too.
# I said that already.
```

Following triple-quoted string is also ignored by Python interpreter and can be used as a multiline comments:

```
...
This is a multiline
comment.
...  
...
```

Chapter 2: Using Variables in Python

2.1 Data Types and Variables

2.1.1 Assigning Values to Variables

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable. For example –

```
counter = 100          # An integer assignment
miles   = 1000.0       # A floating point
name    = "John"        # A string
```

2.1.2 Multiple Assignments

Python allows you to assign a single value to several variables simultaneously. For example –

```
a,b,c = 1,2,"john"
```

Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

2.2 Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types –

- Numbers
- String
- List
- Tuple
- Dictionary

2.2.1 Numeric Data Types

Number data types store numeric values. Number objects are created when you assign a value to them. For example –

```
var1 = 1  
var2 = 10
```

Python supports four different numerical types –

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values)
- complex (complex numbers)

A complex number consists of an ordered pair of real floating-point numbers denoted by $x + yj$, where x and y are the real numbers and j is the imaginary unit.

2.2.2 String Data Type

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator. For example –

```
str = 'Hello World!'  
print str          # Prints complete string  
print str[0]        # Prints first character of the string  
print str[2:5]      # Prints characters starting from 3rd to 5th  
print str[2:]        # Prints string starting from 3rd character  
print str * 2        # Prints string two times  
print str + "TEST" # Prints concatenated string
```

2.2.3 List Data Type

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator. For example –

```

list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print list          # Prints complete list
print list[0]       # Prints first element of the list
print list[1:3]     # Prints elements starting from 2nd till 3rd
print list[2:]      # Prints elements starting from 3rd element
print tinylist * 2  # Prints list two times
print list + tinylist # Prints concatenated lists

```

2.2.4 Tuple Data Type

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as read-only lists. For example –

```

tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')

print tuple          # Prints complete list
print tuple[0]       # Prints first element of the list
print tuple[1:3]     # Prints elements starting from 2nd till 3rd
print tuple[2:]      # Prints elements starting from 3rd element
print tinytuple * 2  # Prints list two times
print tuple + tinytuple # Prints concatenated lists

```

Updation in Tuple is Not-Allowed.

2.2.5 Dictionary Data Type

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]). For example –

```

dict = {}
dict['one'] = "This is one"
dict[2]      = "This is two"

tinydict = {'name': 'john', 'code':6734, 'dept': 'sales'}

```

```

print dict['one']           # Prints value for 'one' key
print dict[2]               # Prints value for 2 key
print tinydict              # Prints complete dictionary

```

Dictionaries have no concept of order among elements. It is incorrect to say that the elements are "out of order"; they are simply unordered.

2.3 Data Type Conversions

Sometimes, you may need to perform conversions between the built-in types. To convert between types, you simply use the type name as a function.

There are several built-in functions to perform conversion from one data type to another. These functions return a new object representing the converted value.

Sr.No.	Function & Description
1	int(x [,base]) – Converts x to an integer. base specifies the base if x is a string.
2	long(x [,base]) – Converts x to a long integer. base specifies the base if x is a string.
3	float(x) – Converts x to a floating-point number.
4	str(x) – Converts object x to a string representation.
5	tuple(s) – Converts s to a tuple.
6	list(s) – Converts s to a list.
7	set(s) – Converts s to a set.
8	dict(d) – Creates a dictionary. d must be a sequence of (key,value) tuples.

Table 2.1 – Data Type Conversion Functions

Chapter 3: Basics of Programming in Python

3.1 Basic Operators

Operators are the constructs which can manipulate the value of operands.

Consider the expression $4 + 5 = 9$. Here, 4 and 5 are called operands and + is called operator.

Types of Operator

Python language supports the following types of operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

3.2 Decision Making

Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions.

Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome is TRUE or FALSE otherwise.

Following is the general form of a typical decision making structure found in most of the programming languages –

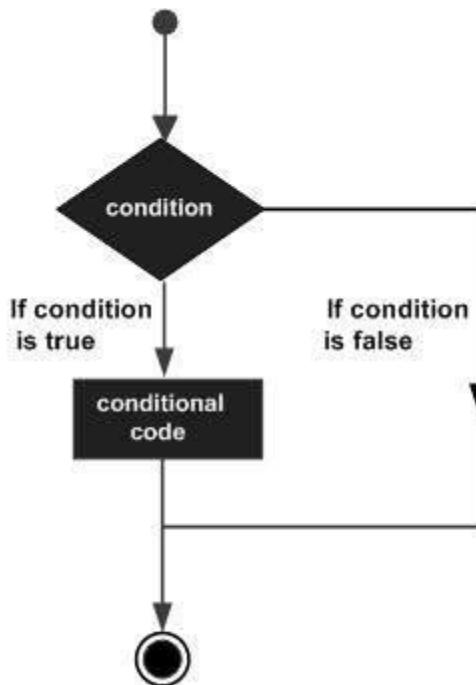


Fig 3.1 – If Condition

Python programming language assumes any non-zero and non-null values as TRUE, and if it is either zero or null, then it is assumed as FALSE value.

Python programming language provides following types of decision making statements.

- if statements – An if statement consists of a boolean expression followed by one or more statements.
- if...else statements – An if statement can be followed by an optional else statement, which executes when the boolean expression is FALSE.
- nested if statements – You can use one if or else if statement inside another if or else if statement(s).

3.3 Loops in Python

In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times. The following diagram illustrates a loop statement –

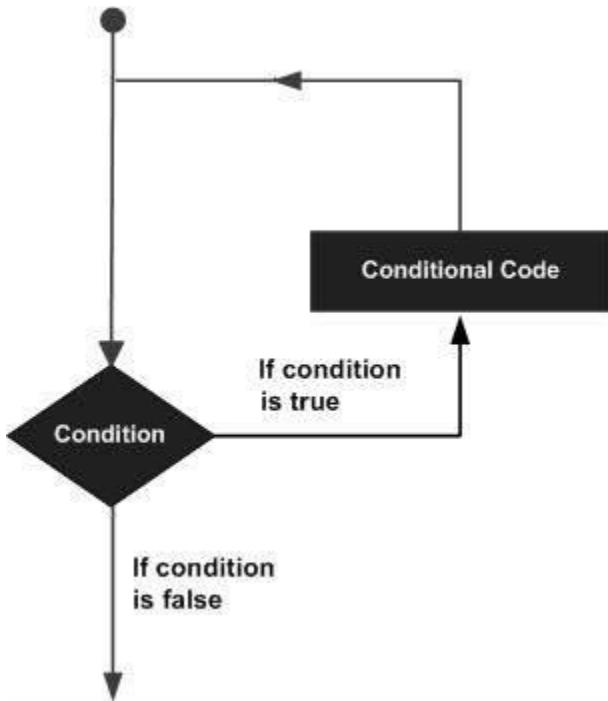


Fig 3.2 – Loops

Python programming language provides following types of loops to handle looping requirements.

- while loop – Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
- for loop – Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
- nested loops – You can use one or more loop inside any another while, for or do..while loop.

3.3.1 Control Statements in Python

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Python supports the following control statements.

- break statement – Terminates the loop statement and transfers execution to the statement immediately following the loop.
- continue statement – Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
- pass statement – The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

3.4 Functions in Python

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

As you already know, Python gives you many built-in functions like `print()`, etc. but you can also create your own functions. These functions are called *user-defined functions*.

3.4.1 Declaring a Function

You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword `def` followed by the function name and parentheses `()`.
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or *docstring*.
- The code block within every function starts with a colon `:` and is indented.
- The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.

Syntax

```
def functionname( parameters ):
    "function_docstring"
    function_suite
    return [expression]
```

3.5 Function Arguments

You can call a function by using the following types of formal arguments –

- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments

3.5.1 Required Arguments

Required arguments are the arguments passed to a function in correct positional order. Here, the number of arguments in the function call should match exactly with the function definition.

3.5.2 Keyword Arguments

Keyword arguments are related to the function calls. When you use keyword arguments in a function call, the caller identifies the arguments by the parameter name.

This allows you to skip arguments or place them out of order because the Python interpreter is able to use the keywords provided to match the values with parameters.

3.5.3 Default Arguments

A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.

3.5.4 Variable Arguments

You may need to process a function for more arguments than you specified while defining the function. These arguments are called *variable-length* arguments and are not named in the function definition, unlike required and default arguments.

3.5.5 The Anonymous Function

These functions are called anonymous because they are not declared in the standard manner by using the *def* keyword. You can use the *lambda* keyword to create small anonymous functions.

- Lambda forms can take any number of arguments but return just one value in the form of an expression. They cannot contain commands or multiple expressions.
- An anonymous function cannot be a direct call to print because lambda requires an expression
- Lambda functions have their own local namespace and cannot access variables other than those in their parameter list and those in the global namespace.
- Although it appears that lambda's are a one-line version of a function, they are not equivalent to inline statements in C or C++, whose purpose is by passing function stack allocation during invocation for performance reasons.

3.5.6 The return Statement

The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.

All the above examples are not returning any value. You can return a value from a function as follows –

```
# Function definition is here

def sum( arg1, arg2 ):

    # Add both the parameters and return them.

    total = arg1 + arg2

    print "Inside the function : ", total

    return total;
```

3.6 Scope of Variables

All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable.

The scope of a variable determines the portion of the program where you can access a particular identifier. There are two basic scopes of variables in Python –

- Global variables
- Local variables

Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.

This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions. When you call a function, the variables declared inside it are brought into scope.

Chapter 4: OOPS in Python

4.1 OOPS Concept

Like other general purpose languages, python is also an object-oriented language since its beginning. Python is an object-oriented programming language. It allows us to develop applications using an Object Oriented approach. In Python, we can easily create and use classes and objects.

Major principles of object-oriented programming system are given below.

- Object
- Class
- Method
- Inheritance
- Polymorphism
- Data Abstraction
- Encapsulation

4.1.1 Object

The object is an entity that has state and behavior. It may be any real-world object like the mouse, keyboard, chair, table, pen, etc.

Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute `__doc__`, which returns the doc string defined in the function source code.

4.1.2 Class

The class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class then it should contain an attribute and method, i.e. an email id, name, age, salary, etc.

Syntax
`class ClassName:
 <statement-1>
 .
 .
 <statement-N>`

4.1.3 Method

The method is a function that is associated with an object. In Python, a method is not unique to class instances. Any object type can have methods.

4.1.4 Inheritance

Inheritance is the most important aspect of object-oriented programming which simulates the real world concept of inheritance. It specifies that the child object acquires all the properties and behaviors of the parent object.

By using inheritance, we can create a class which uses all the properties and behavior of another class. The new class is known as a derived class or child class, and the one whose properties are acquired is known as a base class or parent class.

It provides re-usability of the code.

4.1.5 Polymorphism

Polymorphism contains two words "poly" and "morphs". Poly means many and Morphs means form, shape. By polymorphism, we understand that one task can be performed in different ways. For example You have a class animal, and all animals speak. But they speak differently. Here, the "speak" behavior is polymorphic in the sense and depends on the animal. So, the abstract "animal" concept does not actually "speak", but specific animals (like dogs and cats) have a concrete implementation of the action "speak".

4.1.6 Encapsulation

Encapsulation is also an important aspect of object-oriented programming. It is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.

4.1.7 Data Abstraction

Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonym because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things so that the name captures the core of what a function or a whole program does.

4.2 Python Class and Objects

As we have already discussed, a class is a virtual entity and can be seen as a blueprint of an object. The class came into existence when it instantiated. Let's understand it by an example.

Suppose a class is a prototype of a building. A building contains all the details about the floor, doors, windows, etc. we can make as many buildings as we want, based on these details. Hence, the building can be seen as a class, and we can create as many objects of this class.

On the other hand, the object is the instance of a class. The process of creating an object can be called as instantiation.

4.3 Python Constructor

A constructor is a special type of method (function) which is used to initialize the instance members of the class.

Constructors can be of two types.

1. Parameterized Constructor
2. Non-parameterized Constructor

Constructor definition is executed when we create the object of this class. Constructors also verify that there are enough resources for the object to perform any start-up task.

Chapter 5: Django Framework

It is a python-based open-source back-end web framework. A web framework is a set of components that helps in developing websites/webapps faster. Using Django, we will be creating the back-end of our application/software.

5.1 Introduction

- Why Django? – With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
- Ridiculously fast. Django was designed to help developers take applications from concept to completion as quickly as possible.
- Fully loaded – Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks right out of the box.
- Reassuringly secure – Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.
- Exceedingly scalable – Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.
- Incredibly versatile – Companies, organizations and governments have used Django to build all sorts of things from content management systems to social networks to scientific computing platforms.

5.2 Installing Django

5.2.1 Install Python

Being a Python Web framework, Django requires Python. See What Python version can I use with Django? for details. Python includes a lightweight database called SQLite so you won't need to set up a database just yet.

Get the latest version of Python at <https://www.python.org/downloads/> or with your operating system's package manager.

You can verify that Python is installed by typing `python` from your shell; you should see something like:

```
Python 3.x.y
[GCC 4.x] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

5.2.2 Install Django

You've got three easy options to install Django:

Install an official release. This is the best approach for most users.

Install a version of Django provided by your operating system distribution.

Install the latest development version. This option is for enthusiasts who want the latest-and-greatest features and aren't afraid of running brand new code. You might encounter new bugs in the development version, but reporting them helps the development of Django. Also, releases of third-party packages are less likely to be compatible with the development version than with the latest stable release.

5.2.3 Verifying

To verify that Django can be seen by Python, type `python` from your shell. Then at the Python prompt, try to import Django:

```
>>> import django
>>> print(django.get_version())
2.2
```

5.3 Getting Started with Django

5.3.1 Object-relational mapper

Define your data models entirely in Python. You get a rich, dynamic database-access API for free — but you can still write SQL if needed.

```
class Band(models.Model):
    """A model of a rock band."""
    name = models.CharField(max_length=200)
    can_rock = models.BooleanField(default=True)

class Member(models.Model):
    """A model of a rock band member."""
    name = models.CharField("Member's name", max_length=200)
    instrument = models.CharField(choices=(
        ('g', "Guitar"),
        ('b', "Bass"),
        ('d', "Drums"),
    ),
    max_length=1
)
band = models.ForeignKey("Band")
```

5.3.2 URLs and views

A clean, elegant URL scheme is an important detail in a high-quality Web application. Django encourages beautiful URL design and doesn't put any cruft in URLs, like .php or .asp.

To design URLs for an application, you create a Python module called a URLconf. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.

This mapping can be as short or as long as needed. It can reference other mappings. And, because it's pure Python code, it can be constructed dynamically.

Django also provides a way to translate URLs according to the active language. See the internationalization documentation for more information.

```
from django.urls import path
from . import views

urlpatterns = [
    path('bands/', views.band_listing, name='band-list'),
    path('bands/<int:band_id>', views.band_detail, name='band-detail'),
    path('bands/search/', views.band_search, name='band-search'),
]

from django.shortcuts import render

def band_listing(request):
    """A view of all bands."""
    bands = models.Band.objects.all()
    return render(request, 'bands/band_listing.html', {'bands': bands})
```

How Django processes a request

- When a user requests a page from your Django-powered site, this is the algorithm the system follows to determine which Python code to execute:
- Django determines the root URLconf module to use. Ordinarily, this is the value of the ROOT_URLCONF setting, but if the incomingHttpRequest object has a urlconf attribute (set by middleware), its value will be used in place of the ROOT_URLCONF setting.
- Django loads that Python module and looks for the variable urlpatterns. This should be a sequence of django.urls.path() and/or django.urls.re_path() instances.
- Django runs through each URL pattern, in order, and stops at the first one that matches the requested URL.

- Once one of the URL patterns matches, Django imports and calls the given view, which is a simple Python function (or a class-based view). The view gets passed the following arguments: An instance of `HttpRequest`.
- If the matched URL pattern returned no named groups, then the matches from the regular expression are provided as positional arguments.
- The keyword arguments are made up of any named parts matched by the path expression, overridden by any arguments specified in the optional `kwargs` argument to `django.urls.path()` or `django.urls.re_path()`.
- If no URL pattern matches, or if an exception is raised during any point in this process, Django invokes an appropriate error-handling view.

5.3.3 Templates

Django's template language is designed to strike a balance between power and ease. It's designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed.

```
<html>
  <head>
    <title>Band Listing</title>
  </head>
  <body>
    <h1>All Bands</h1>
    <ul>
      {% for band in bands %}
        <li>
          <h2><a href="{{ band.get_absolute_url }}">{{ band.name }}</a></h2>
          {% if band.can_rock %}<p>This band can rock!</p>{% endif %}
        </li>
      {% endfor %}
    </ul>
  </body>
</html>
```

5.3.4 Forms

Handling forms is a complex business. Consider Django's admin, where numerous items of data of several different types may need to be prepared for display in a form, rendered as HTML, edited using a convenient interface, returned to the server, validated and cleaned up, and then saved or passed on for further processing.

Django's form functionality can simplify and automate vast portions of this work, and can also do it more securely than most programmers would be able to do in code they wrote themselves.

Django handles three distinct parts of the work involved in forms:

- preparing and restructuring data to make it ready for rendering
- creating HTML forms for the data
- receiving and processing submitted forms and data from the client

It is possible to write code that does all of this manually, but Django can take care of it all for you.

Django provides a powerful form library that handles rendering forms as HTML, validating user-submitted data, and converting that data to native Python types. Django also provides a way to generate forms from your existing models and use those forms to create and update data.

```
from django import forms

class BandContactForm(forms.Form):
    subject = forms.CharField(max_length=100)
    message = forms.CharField()
    sender = forms.EmailField()
    cc_myself = forms.BooleanField(required=False)
```

5.3.5 Authentication

Django comes with a full-featured and secure authentication system. It handles user accounts, groups, permissions and cookie-based user sessions. This lets you easily build sites that let users to create accounts and safely log in/out.

```
from django.contrib.auth.decorators import login_required
from django.shortcuts import render

@login_required
def my_protected_view(request):
    """A view that can only be accessed by logged-in users"""
    return render(request, 'protected.html', {'current_user': request.user})
```

The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is who they claim to be, and authorization determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

The auth system consists of:

- Users
- Permissions: Binary (yes/no) flags designating whether a user may perform a certain task.
- Groups: A generic way of applying labels and permissions to more than one user.
- A configurable password hashing system
- Forms and view tools for logging in users, or restricting content

- A pluggable backend system

The authentication system in Django aims to be very generic and doesn't provide some features commonly found in web authentication systems. Solutions for some of these common problems have been implemented in third-party packages:

- Password strength checking
- Throttling of login attempts
- Authentication against third-parties (OAuth, for example)
- Object-level permissions

5.3.6 Admin

One of the most powerful parts of Django is its automatic admin interface. It reads metadata in your models to provide a powerful and production-ready interface that content producers can immediately use to start managing content on your site. It's easy to set up and provides many hooks for customization.

```
from django.contrib import admin
from bands.models import Band, Member

class MemberAdmin(admin.ModelAdmin):
    """Customize the look of the auto-generated admin for the Member
model"""
    list_display = ('name', 'instrument')
    list_filter = ('band',)

admin.site.register(Band) # Use the default options
admin.site.register(Member, MemberAdmin) # Use the customized options
```

5.3.7 Security

Django provides multiple protections against:

- Clickjacking
- Cross-site scripting
- Cross Site Request Forgery (CSRF)
- SQL injection
- Remote code execution

Chapter 6: Project – College ERP System

6.1 Introduction

Enterprise Resource Planning system, popularly known as ERP system has become very popular as an enterprise management software tool. It was the larger companies that have opted to use the ERP systems initially. However, the use of ERP has changed and today the term can refer to any type of company, no matter what industry it falls in. In fact, ERP systems are used in almost any type of organization - large or small. The latest ERP tools available in the market today can cover a wide range of functions and integrate them into one unified database. This made ERP to land up into higher educational institutes. In today's competitive business world usage of ERP system is becoming a must for any educational organization to meet the challenges faced in their business process and to have a cutting edge. So in order to be different and ready for action the institutes need a central resource planning that can manage the entire information and operations of the institutions.

6.1.1 Identification

The software system being considered for development is referred to as College ERP System. The ultimate customer, or end-user, of the system will be the teachers and the HoDs of each department in college, Students and Administrators.

6.1.2 Purpose

The purpose is to design software for college database which contains up to date or accurate information of the college. That should improve efficiency and flexibility of college record management and to provide a common and or simple platform for everyone to access the student's information. College Automation System consists of different modules such as student, faculty, admin etc. Our main purpose is to create a software which will manage the working of these different modules. The interconnectivity among modules reduces the time to perform different operational task.

6.1.3 Scope

College management is becoming a very essential component in education in this modern day age. With the help of College Automation System we can gather all the useful information needed to the management in few clicks. The College ERP system now computerizes all the details that are maintained manually. Once the details are fed into the system or computer there is no need for various persons to deal with separate sections. Only a person is enough to maintain all the reports and records. The security can also be given as per the user requirement

6.1.4 Definitions, Acronyms and Abbreviations

- CIE – Continuous Internal Evaluation, series of examinations conducted throughout the semester to assess the academic performance of the student. CIE is a total of 50 marks
- SEE – Semester End Examination, conducted at the end of each semester to assess the academic performance of the student. Conducted for 50 marks.

6.2 Overall Description

6.2.1 Product Perspective

ERP means the techniques and concepts for integrated management of business as a whole, from the viewpoint of effective use of management resources to improve the efficiency of enterprise management. A fully integrated web-based ERP will capture and create accurate, consistent and timely relevant data, and assist in intelligent business decision-making. The primary purpose of E-college is to provide mechanisms for automated processing and management of the entire institution. It reduces data error, ensures that information is managed efficiently and is always up-to-date. Complete student histories for all years, can easily be searched, viewed and reported on press of button.

It is made after extensive study of all the departments like student, faculty, etc of colleges and is provided with the extract of everything a college requires for their database handling, department management and student/staff management. The security issue within ERP has been there for a long time, but most of the solutions are based on the assumption that an ERP system is a closed environment.

6.2.2 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 (still being implemented).
- The students will be able to Communicate and provide feedback to their teachers (still being implemented).
- The students will have access to a forum page where they can communicate with each other (still being implemented).
- The administrator will be able to view and update information such as departments, classes, teachers, students, courses.

6.2.3 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.

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.

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.

The administrator should have the following features:

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

6.3 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 incorporates information hiding. The software is divided into 3 modules which are students, teachers and administrators. We shall look at each module in detail.

6.3.1 Student

Each student belongs to a class identified by semester and section. Each class belongs to a department and are 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 if below a specified threshold, say 75%, It will be marked in red otherwise it 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 calender 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. Notification are messages from the admin such as declaration of holidays, test time-table etc. The events and their details are specified here.

6.3.2 Teacher

Each teacher belongs to a department and are 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 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 card including all the marks and CGPA of a student can be generated.

6.3.3 Administrator

The administrator will have access to all the information in the different tables in the database. They will access to all the tables in a list form. They will be able to add a 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.

6.3.4 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

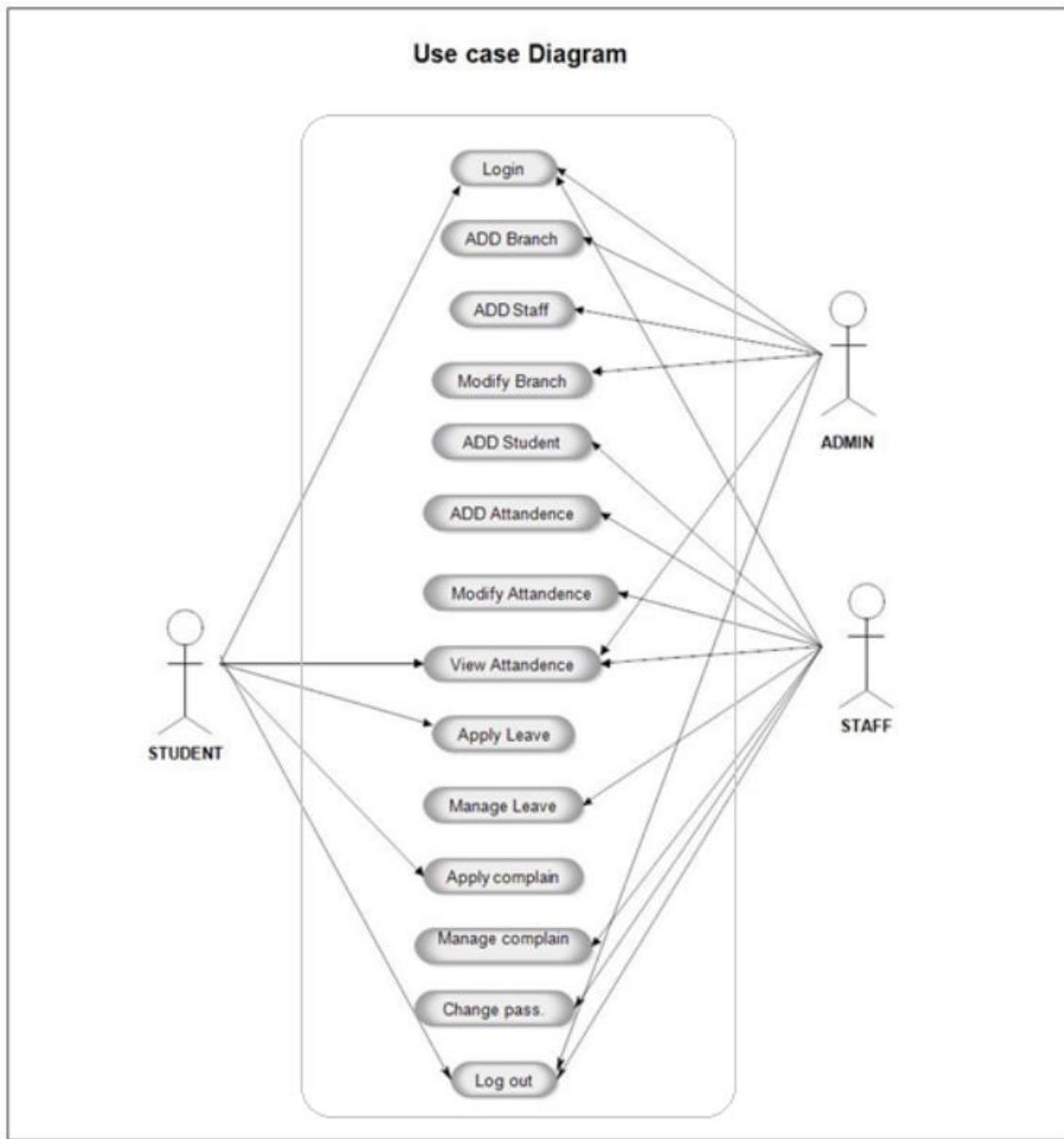


Fig 6.1 – Use Case Diagram of College ERP System

6.3.5 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

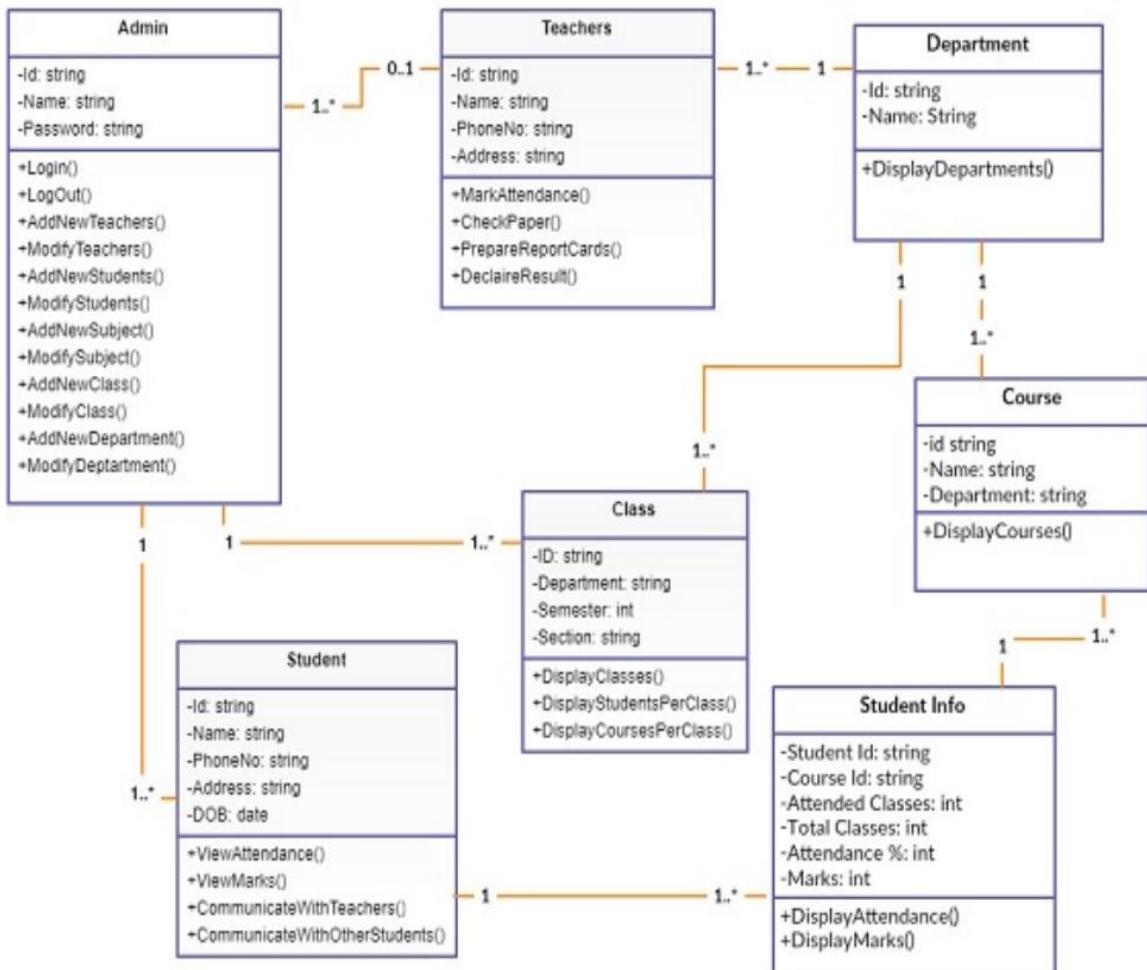


Fig 6.2 – Class Diagram of College ERP System

6.3.6 Entity Relationship Diagram

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

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.

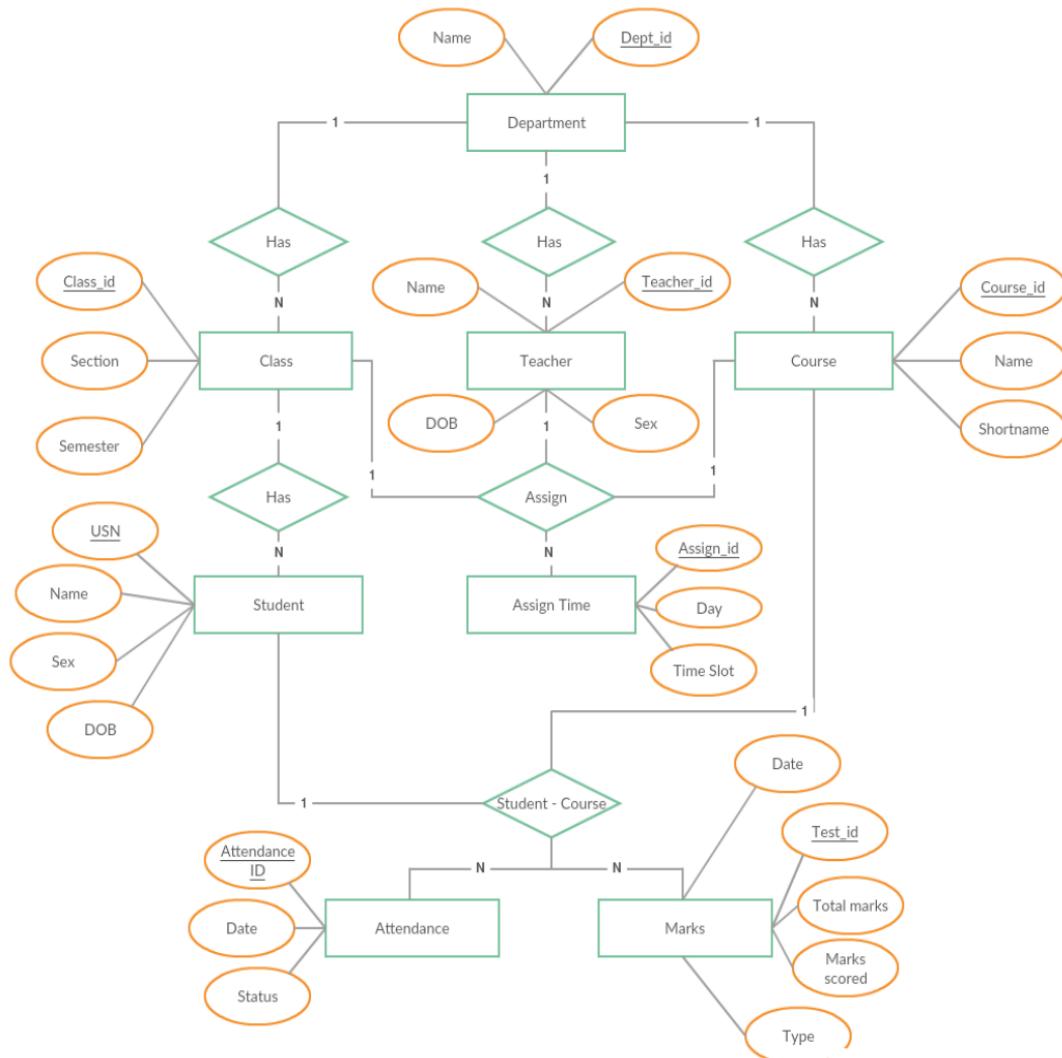


Fig 6.3 – ER Diagram of College ERP System

6.4 Modules in the System

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.

6.4.1 Student

Login

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

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 timetable provides the classes assigned to that student and day and time of each in a tabular form.

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 display 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.

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.

Time Table

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 assign table, which is a table containing the information of all the teachers assigned to a class with a course and the timings the classes.

6.4.2 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.

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.

Attendance

There is a list of all the class assigned to teacher. So, for each class there are 3 actions available. They are:

- Enter Attendance – On this page, the classes scheduled or conducted is listed in the form of a list. Initially, all the scheduled classes will be listed from the start of the semester to the current date. Thus, if there is class scheduled for today, it will automatically appear on top of the list. If the attendance of any day is not marked it will be red, otherwise green if marked. Classes can also be cancelled which will make that date as yellow. While entering the attendance, the list of students in that class is listed and there are two options next to each. These options are in the form of a radio button for present and absent. All the buttons are initially marked as present and the teacher just needs to change for the absent students.
- Edit Attendance – After entering attendance, the teacher can also edit it. It is similar to screen for entering attendance, only the entered attendance is saved and display. The teacher can change the appropriate attendance and save it.
- Extra Class – If a teacher has taken a class other than at the scheduled timings, they may enter the attendance for that as well. While entering the extra class, the teacher just needs to specify the date it was conducted and enter the attendance of each of

the students. After submitting extra class, it will appear in the list of conducted classes and thus, it can be edited.

- Student Attendance – For each assigned class, the teacher can view the attendance status of the list of students. The number of attended classes, total number of classes conducted and the attendance percentage is displayed. If the attendance percentage of any of the students is below 75, it will be displayed in red. Thus, the teacher may easily find the list of students not eligible to take a test.
- 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.

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 if there is a need to change the marks of any student, it is possible to edit the 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 change immediately.
- Student Marks – For each assigned class, the teacher has access to the list of students and the marks they obtained in all the tests. This is displayed in a tabular form.
- Time Table – This page is a table which lists the day and timings of each of the classes assigned to the teacher. 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 assign table, which is a table containing the information of all the teachers assigned to a class with a course and the timings the classes.
- Free Teachers – For each entry in the table, the list of free teachers can be generated. Free teachers are the teachers who assigned to the class and are free for that time slot on that day. This is very useful for the teachers particularly when they are on leave as it helps them find a suitable replacement for that class.
- Reports – The last page for the teachers is used to generate reports for each class. The report specifies the list of students in that class and their respective CIE and attendance percentage. CIE is the average of the marks obtained from the tests, 3

internals and 2 events. The CIE is out of 50 and the students with CIE below 25 are marked in red and are not eligible to write the semester end exam. Also, the attendance percentage is displayed with students below 75% marked in red.

6.4.3 Administrator

The administrator is responsible for adding and maintaining all the departments, students, teachers, classes and courses. All this data is stored in the database in their respective tables. The admin is also responsible for adding and maintaining the list of teachers assigned to class with a course and the timings. This information is stored in the Assign table. The admin also has access to the marks and attendance of each student and can modify them.

There are several features in place to ensure that querying the database is quick and efficient for the administrator. As the database has the potential to become huge, there is a search feature for every table including student, teacher etc. The search has get a specific record based on name or id. Also, it can filter the record based on department, class etc.

6.5 Screenshots of the Implemented System

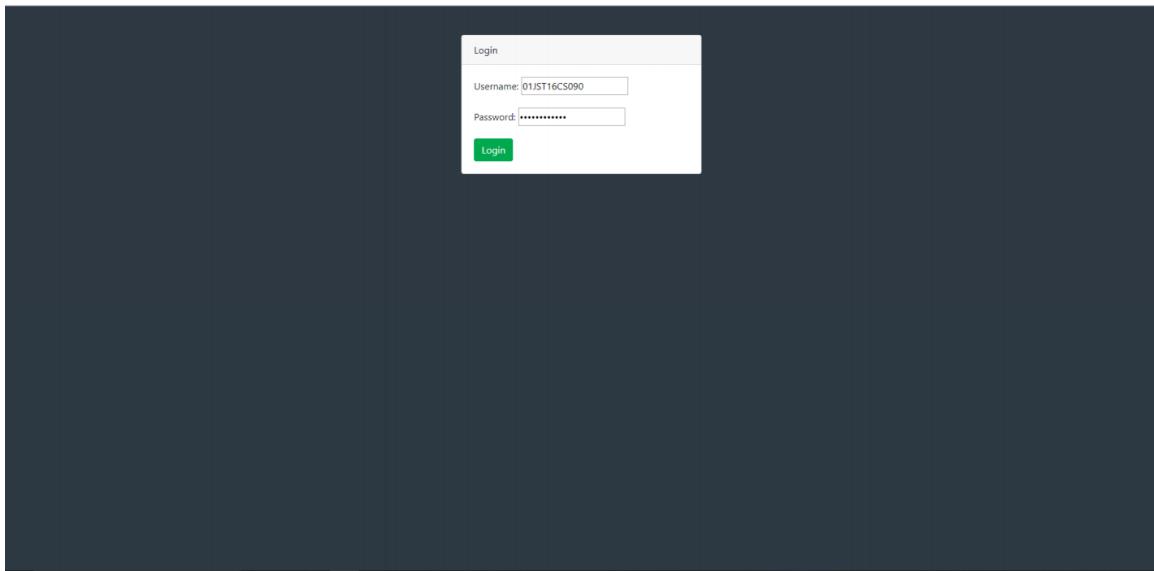


Fig 6.4 – Login Page

A screenshot of the student home page for "CollegeERP". The top navigation bar includes the college name and a "Logout" link. The main content area features a welcome message "Welcome Samarth" and three cards: "Attendance" (with a clipboard icon), "Marks" (with a grade icon), and "TimeTable" (with a calendar icon). Each card has a brief description and a "View" button.

Fig 6.5 – Student Home Page

CollegeERP ☰

Samarth Logout

Attendance

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

Fig 6.6 – Student Attendance Page

CollegeERP ☰

Samarth Logout

Software Engineering

#	Date	Day	Status
1	Oct. 26, 2018	Friday	Present
2	Oct. 29, 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

Fig 6.7 – Student Attendance Detail Page

The screenshot shows a student marks page from a CollegeERP system. The top navigation bar includes 'CollegeERP' and 'Logout'. On the left sidebar, there are links for 'Home', 'Attendance', 'Attendance By Subject', 'Marks', and 'Time Table'. The main content area is titled 'Marks' and displays a table with the following data:

Course ID	Course name	Internals 1	Internals 2	Internals 3	Event 1	Event 2	SEE
CS510	Database Management System	0	0	0	0	0	0
CS520	UNIX	0	0	0	0	0	0
CS530	Software Engineering	9	15	10	15	15	0
CS540	Computer Networks	0	0	0	0	0	0
CS550	Language Processor	0	0	0	0	0	0
MA510	Linear Algebra	0	0	0	0	0	0

Fig 6.8 – Student Marks Page

The screenshot shows a student time table page from a CollegeERP system. The top navigation bar includes 'CollegeERP' and 'Logout'. On the left sidebar, there are links for 'Home', 'Attendance', 'Attendance By Subject', 'Marks', and 'Time Table'. The main content area is titled 'Timetable' and displays a weekly grid from Monday to Saturday. The columns represent time slots from 7:30 AM to 4:30 PM. The rows represent days of the week. The grid shows course assignments for each slot, such as CS540, CS510, CS530, etc.

	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					CS540	CS510	CS530	CS550			
Tuesday						MA510	CS510	CS540	CS520	CS550	
Wednesday						MA510	CS520	CS530			
Thursday									CS540	CS530	CS510
Friday					MA510	CS520	CS550	CS530			
Saturday		MA510	CS510		CS540	CS520	CS550				

Fig 6.9 – Student Time Table Page

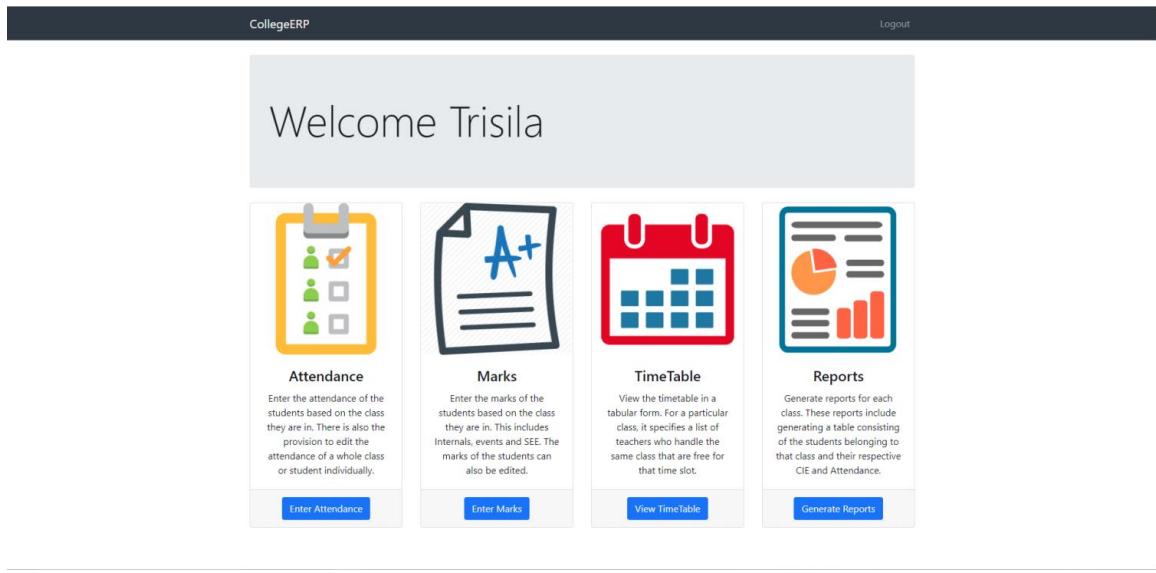


Fig 6.10 – Teacher Home Page

CollegeERP		Trisila Logout
	Home	
	Attendance	
	Dakshath	Present Absent
	Farhan	Present Absent
	Nihal	Present Absent
	Nikhil	Present Absent
	Rajat	Present Absent
	Renu	Present Absent
	Anudeep	Present Absent
	Samarth	Present Absent
	Shahid	Present Absent
	Shravan	Present Absent
	Boss	Present Absent
	Tejus	Present Absent

Fig 6.11 – Entering Attendance

Student name	
Dakshath	[Present] [Absent]
Farhan	[Present] [Absent]
Nihal	[Present] [Absent]
Nikhil	[Present] [Absent]
Rajat	[Present] [Absent]
Renu	[Present] [Absent]
Anudeep	[Present] [Absent]
Samarth	[Present] [Absent]
Shahid	[Present] [Absent]
Shravan	[Present] [Absent]
Boss	[Present] [Absent]
Tejus	[Present] [Absent]

Fig 6.12 – Editing Attendance

Attendance					
USN	Student name	Attended classes	Total classes	Attendance %	Classes to attend
CS01	Dakshath	8	10	80.0	0
CS02	Farhan	10	10	100.0	0
CS03	Nihal	7	10	70.0	2
CS04	Nikhil	8	10	80.0	0
CS05	Rajat	10	10	100.0	0
CS06	Renu	10	10	100.0	0
CS07	Anudeep	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	2
CS11	Boss	10	10	100.0	0
CS12	Tejus	9	10	90.0	0
CS13	Vijeth	8	10	80.0	0

Fig 6.13 – Attendance of Student in Class

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

Fig 6.14 – Attendance Details of an individual Student

Student Name	Total Marks	Enter Marks
Dakshath	20	20
Farhan	20	15
Nihal	20	12
Nikhil	20	19
Rajat	20	14
Renu	20	12
Anudeep	20	15
Samarth	20	9
Shahid	20	8
Shravan	20	7
Boss	20	15
Tejus	20	14
Vijeth	20	12

Fig 6.15 – Editing Marks

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	19	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	Boss	15	13	20	18	15	0
CS12	Tejas	14	12	10	19	15	0
CS13	Vijeth	12	15	20	20	15	0

Fig 6.16 – Marks of all the Students

Marks			
Student USN	Student Name	Attendance	CIE
CS01	Dakshath	80.0	35
CS02	Farhan	100.0	35
CS03	Nihal	70.0	40
CS04	Nikhil	80.0	38
CS05	Rajat	100.0	40
CS06	Renu	100.0	31
CS07	Anudeep	80.0	37
CS08	Samarth	80.0	32
CS09	Shahid	100.0	37
CS10	Shravan	70.0	24
CS11	Boss	100.0	41
CS12	Tejas	90.0	35
CS13	Vijeth	80.0	41

Fig 6.17 – CIE and Attendance of the class

6.6 Conclusion

By using Existing System accessing information from files is a difficult task and there is no quick and easy way to keep the records of students and staff. Lack of automation is also there in the Existing System. The aim of Our System is to reduce the workload and to save significant staff time. Title of the project as College ERP System is the system that deals with the issues related to a particular institution. It is the very useful to the student as well as the faculties to easy access to finding the details. The college ERP provides appropriate information to users based on their profiles and role in the system. This project is designed keeping in view the day to day problems faced by a college system. The fundamental problem in maintaining and managing the work by the administrator is hence overcome. Prior to this it was a bit difficult for maintaining the time table and also keeping track of the daily schedule. But by developing this web-based application the administrator can enjoy the task, doing it with ease and also by saving the valuable time. The amount of time consumption is reduced and also the manual calculations are omitted, the reports can be obtained regularly and also whenever on demand by the user. The effective utilization of the work, by proper sharing it and by providing the accurate results. The storage facility will ease the job of the operator. Thus the system developed will be helpful to the administrator by easing his/her task. This System provide the automate admissions no manual processing is required. This is a paperless work. It can be monitored and controlled remotely. It reduces the man power required. It provides accurate information always. All years together gathered information can be saved and can be accessed at any time. The data which is stored in the repository helps in taking intelligent decisions by the management providing the accurate results. The storage facility will ease the job of the operator. Thus the system developed will be helpful to the administrator by easing his/her task providing the accurate results. The storage facility will ease the job of the operator. This project is successfully implemented with all the features and modules of the college management system as per requirements.

References

1. <https://en.wikipedia.org/wiki/Requirements-engineering>
2. <https://web.cs.dal.ca/hawkey/3130/srs-template-ieee.doc>
3. https://en.wikipedia.org/wiki/Class_diagram
4. <https://www.djangoproject.com/>
5. <https://getbootstrap.com/>
6. <https://www.tutorialspoint.com/>
7. <https://creately.com/>
8. <https://www.overleaf.com/project>