COLLEGE MANAGEMENT SYSTEM

A Project Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Rohan Bag (Roll No 01)

&

Ankit Meshram (Roll No 17)

Under the esteemed guidance of

Mr. Agi Thomas

Head of Department

DEPARTMENT OF INFORMATION TECHNOLOGY

SMT.CHANDIBAI HIMATHMAL MANSUKHANI COLLEGE

(Affiliated to University of Mumbai)

ULHASNAGAR, 421003

MAHARASHTRA

2021-22

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PRN No.: …………………… **Roll no: _01&17__**

1. Name of the Student

Rohan Bag &Ankit Meshram

2. Title of the Project

_College Management System

3. Name of the Guide

_Mr. Agi Thomas

4. Teaching experience of the Guide 19 years

5. Is this your first submission? **Yes □** **No □**

Signature of the Student **Signature of the Guide**

Date: ………………… **Date:**

…………………….

Signature of the HOD

Date: …………………

SMT. CHANDIBAI HIMATHMAL MANSUKHANI COLLEGE

(Affiliated to University of Mumbai)

ULHASNAGAR-MAHARASHTRA- 421003

DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

This is to certify that the project entitled **College Management System** is a bonafied work of **Rohan Bag &Ankit Meshram** bearing Seat.No: 01 & 17 submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Head, B.Sc. (IT)**

External Examiner

Date: **College Seal**

ABSTRACT

In the Modern Era of fast developing technology, we can do things which we wouldn't have imagined of and to accomplish these we would need a platform that would ease and automate our tasks with comfort. The purpose of our College Management System is to automate the existing manual system by the help of mostly automatic web application fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Thus reduce the manual work for managing the College, Faculty, Student, Course. The required software and hardware are easily available and easy to work with. The main aim of our project is to simplify for the users (Students, Faculty Members, Teachers, Administrative). It is a simplified web application with user interactive GUI. The administrator of faculty can direct access and have all updates he may need. Our project prioritizes for least human interaction possible for the administration. The purpose of the project is to build an application program to reduce the manual work for managing the College, Faculty, Student, Course. It also negates all human error possible. It has various basic attributes such as reminders via emails/messages, notices can be posted by the administration and have various levels of access in messages, notices, this program can change the way of interaction between the system and the end users. This application will aid the user by saving a lot of time by automation. Although the application would provide a lot of automated functions, the overall functions are admin centric and support the admin’s preferences.

ACKNOWLEDGEMENT

In preparation of our assignment, we had to take the help and guidance of some respected persons, who deserve our deepest gratitude. As the completion of this assignment gave us much pleasure, we would like to show our gratitude **Mr. Agi Thomas**, Course Instructor, on Mumbai University for giving us the good guidelines for documentation throughout numerous consultations. We would also like to expand our gratitude to all those who have directly and indirectly guided us in writing this assignment.

In addition, we would like thank to Professor **Mr. Agi Thomas**, who introduced us to the Methodology of work, and whose passion for the ―underlying structures‖ had lasting effect. Also a huge thanks to our principal **Dr. Manju Lalwani Pathak** for providing us all the resources required for this documentation.

We would also like to thank other teaching and non-teaching staff who in some or other way directly or indirectly helped us and guided us for this.

Many people, especially our classmates have made valuable comment suggestions on this proposal which gave us an inspiration to improve our documentation. We thank all the people for their help directly and indirectly to complete our documentation.

DECLARATION

I hereby declare that the project entitled, "**College Management System**" done at **Smt.CHM College - Ulhasnagar**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION		10
	1.1	Background	10
	1.2	Objectives	11
	1.3	Purpose, Scope, and Applicability	12
		1.3.1 Purpose	12
		1.3.2 Scope	12
		1.3.3 Applicability	13
	1.4	Achievements	
	1.5	Organisation of Report	
CHAPTER 2	SURVEY OF TECHNOLOGIES		14
CHAPTER 3	REQUIREMENTS AND ANALYSIS		23
	3.1	Problem Definition	23
	3.2	Requirements Specification	24
	3.3	Planning and Scheduling	24
	3.4	Software and Hardware Requirements	27
	3.5	Preliminary Product Description	33
	3.6	Conceptual Models	33
CHAPTER 4	SYSTEM DESIGN		55
	4.1	Basic Modules	55
	4.2	Data Design	55
		4.2.1 Schema Design	57
		4.2.2 Data Integrity and Constraints	60
	4.3	Procedural Design	65
		4.3.1 Logic Diagrams	66
		4.3.2 Data Structures	68
		4.3.3 Algorithms Design	68
	4.4	User interface design	70
	4.5	Security Issues	72
	4.6	Test Cases Design	73

CHAPTER 5	IMPLEMENTATION AND TESTING		75
	5..1	Implementation Approaches	75
	5.2	Coding Details and Code Efficiency	76
		5.2.1 Coding Details	76
		5.2.2 Code Efficiency:	104
	5.3	Testing Approach:	105
		5.3.1 Unit Testing:	105
		5.3.2 Integrated Testing:	105
		5.3.3 Beta Testing:	105
	5.4	Modifications and Improvement:	107
	5.5	Test Cases:	107
CHAPTER 6	RESULTS AND DISCUSSION		111
	6.1	Test Reports	111
	6.2	User Documentation:	113
CHAPTER 7	CONCLUSIONS		124
	7.1	Conclusion:	124
		7.1.1 Significance of the System:	124
	7.2	Limitations of the System:	124
	7.3	Future scope of the Project:	125
		References & links	126

List of Figures

Figure 3-1	SDLC	23
Figure 3-2	Gantt Char Table	**29**
Figure 3-3	Gantt Char	**30**
Figure 3-4	Pert Chart	**31**
Figure 3-5	ER Diagram	**34**
Figure 3-6	Use case Diagram	**40**
Figure 3-7	Sequence Diagram	**45**
Figure 3-8	Data Flow diagram	**48**
Figure 3-9	System Flowchart	**50**
Figure 4-1	Class Diagram	**53**
Figure 4-2	DBMS	**56**
Figure 4-3	Schema Design	**59**
Figure 4-4	Logical Diagrams	**66**
Figure 4-5	UI	**70**
Figure 3-1	Test cases	**73**

Chapter 1

Introduction

Android, Apple, Windows, etc. are various famous and commonly used Operating Systems. All the Operating Systems provides plenty of applications and services for users. For an instance, the Mail is used to store and send mails across the globe from one user to another irrespective of various devices. Similarly, our project can work on any platform having a web browser. Our project College management system is an integrated web application that handles various academic and non-academic activities of a College/Academic Institute. The system can access by every students/faculties/employees of the institution through internet connected computers or internet enabled mobile devices with the aid of his username and password. It can assist the administrator faculty to concentrate on their other activities rather than to maintain various records. Using this system you can manage all college management work like admission, fees submission, time table management, assistance for conducting examination, posting notices, generating report cards, result declaration, etc. Using this system you can manage all information of all aspects of a college, its students, faculties, Departments, marks and other curricular activities. College management system provides the easiest way to manage all functionalities of a college. This system facilitates colleges to maintain the functionality related to college employees and their students.

1.1 BACKGROUND

Nowadays the technology has been vastly developed so our primary focus is the User Experience and access to the application easily from any geolocation. The basic idea of the project is to create an application which helps less tech savvy to operate the system with ease. Our proposed system has the requirement of Internet Connectivity. We have many features in our system which helps the user to ease tasks. Thus, these various features give birth to various functionalities to be implemented in systems. Our application will help the user to ease these functionalities.

1.2 OBJECTIVE

The main objective of college management system is to automate all functionalities of a college or university. Using this system you can manage all college management work like admission, fees submission, time table management and result declaration. Using this college management system you can view or update data and information about students and staff easily. This system helps in managing the activity like student admission, student registration, fees submission. Admin can also retrieve information of employee student.

The COLLEGE MANAGEMENT SYSTEM can be used to store student information like attendance, fees, and student result etc. admin can create report regarding any student any time using this system. Using this system you can register new student and their course details. You can submit students fees and can check fees details anytime. You can create exam result and submit in this system. Student can check their result online by logging to the system. You can also add new employee in the system and can check details of the employee easily. Student can also check course detail online from this system.

Using this system you can manage all information of all aspects of a college, its students, faculties, Departments, marks and other curricular activities. College management system provides the easiest way to manage all functionalities of a college. This system facilitates colleges to maintain the functionality related to college employees and their students.

College Management System can store and manage all data of the various departments of a college like Administration, Attendance, Staff details etc. using this system user can retrieve any information related to student, teacher and fees. Using this system teacher can check student attendance anytime. This system also help teacher to announce the result. College administration can also manage college work easily. Admin can check leave, salary and other details of teacher any time. They can also create time table of classes from this system. The Library module is used for the data process of library and book accessing for students and staffs.

1.3 PURPOSE , SCOPE & APPLICABILITY

1.3.1 PURPOSE

This is a web-oriented application allows us to access the whole information about the college, staffs, students, facilities etc. This application provides a virtual tour of Campus. Here we will get the latest information about the students and staffs. This generic application designed for assisting the students of an institute regarding information on the courses, subjects, classes, assignments, grades and timetable. It also provides support that a faculty can also check about his daily schedule, can upload assignments, and notices to the students. Here administrator will manage the accounts of the student and faculties, makes the timetable, and upload the latest information about the campus.

1.2.2 SCOPE

- College information: Through this service one can access the complete information about the college campus such as courses available, admission procedure, placements, college events, achievements etc.

- Student attendance status: It gives the attendance status of students. Faculty will update the attendance periodically and can be seen by students and parents.

- Student's performance in exams: This facility provides the performance of the student in each exam which is conducted by university or college such as midterm performance. Marks obtained by students in exams will be updated by faculties that can be access by students and parents.

- Exam Notification: This facility notifies students and parents about examination schedule.

- Events: It will give information about different events that will be conducted by college time to time. Information about these events will be updated by administrator.

- Information about staff: It will help in maintaining complete information about college faculty members such as their department, cadre, date of joining, salary, etc.

Administrator will register new faculties and remove their account when they leave the college.

1.3.3 Applicability

Some of the applicability that are provided is:

1. Simplicity: The interface of the application would be simple thus providing users a platform to use computer in easier way.

2. Compatibility: The application is compatible with all devices having the latest browser.

3. Performance: Our application gives a better experience and performance, thus reducing the human efforts on performing tasks.

Chapter 2

SURVEY OF TECHNOLOGIES

2.1 Existing system

The existing system had many shortcomings and hence increased chance of manual errors such as:

- It have a lots of manual work (Manual system does not mean that we are working with pen and paper, it also include working on spread sheets and other simple software's)
- It requires more no of employees need to work.
- It was time consuming process.
- The present system was very less secure.
- It is unable to generate different kinds of report.
- It was less user-friendly.

2.2 Proposed system

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the Internet.
- Various classes have been used to provide file upload and mail features.
- There is no risk of data mismanagement at any level while the project development is under process.
- It provides high level of security using different protocols like https etc.

2.3 Survey of Technologies

AVAILABLE TECHNOLOGIES

Following are the technologies available for this project. There are separate technologies for website.

For website development: .

FRONT END/GUI TOOLS	**.net technologies, java, Html/CSS , bootstrap, react**
DBMS/BACK END	**Oracle, SQL Plus, MySQL, SQL**
LANGUAGES	**C, C++, Java, Python**
SCRIPTING LANGUAGES	**Php, java script, Node js,express**

. .

FRONT END/GUI TOOLS:

1. .Net Technologies

The .NET is nothing but another name of facilities. These facilities can be utilized by beginners, Developers as well as by Expert. The visual Studio .NET is a complete set of development tools for building ASP web applications, XML web services, desktop applications, and mobile applications. VB .NET, C++ .NET, visual C# .NET, and visual j# .NET, all use the same integrated development environment (IDE).

The .NET is the technology from Microsoft, on which all other Microsoft technologies will be depending on in future. It is a major technology change, introduced by Microsoft, to catch the market from the SUN's Java.

2. Java

Java is a general-purpose programming language that is class-based, object-oriented (although not a pure object-oriented language, as it contains primitive types, and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte-code that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications.

3. HTML/CSS

HTML (the Hypertext Mark-up Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications.

4. Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.it focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents.

5.REACT

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies.React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

DBMS/BACK END:

1. Oracle

Oracle is one of the largest vendors in the enterprise IT market and the shorthand name of its flagship product, a relational database management system (RDBMS) that's formally called Oracle Database. The database software sits at the center of many corporate IT environments, supporting a mix of transaction processing, business intelligence and analytics applications.

2. SQL Plus

SQLPlus is an interactive and batch query tool that is installed with every Oracle Database installation. It has a command-line user interface, a Windows Graphical User Interface (GUI) and the iSQLPlus web-based user interface. There is also the SQLPlus Instant Client which is a stand-alone command-line interface available on platforms that support the OCI Instant Client. SQLPlus Instant Client connects to any available Oracle database, but does not require its own Oracle database installation.

3. MySQL

MySQL is an open-source relational database management system (RDBMS).Its name is combinations of ―My‖, the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

4. SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

LANGUAGES:

1. C

C is a general-purpose, imperative computer programming language, and supporting structured programming, lexical variable scope and recursion.C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming.

2. C++

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, and IBM, so it is available on many platforms.

3. Java

Java is a general-purpose computer-programming language that is concurrent, class based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

4.Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

SCRIPTING LANGUAGES

1.Php

PHP is a general-purpose scripting language geared towards web development.It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

2.Javascript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. Most web browsers have a dedicated JavaScript engine to execute the code on the user's device.

3.Node.js

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

4.Express.js

Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. The original author, TJ Holowaychuk, described it as a Sinatra-inspired server, meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

Chosen Technologies

Following are the technologies which have been selected for this project.

For website development:

FRONT END/GUI TOOLS	**Html/CSS , bootstrap, react**
DBMS/BACK END	**MySQL**
LANGUAGES	**Python**
SCRIPTING LANGUAGES	**Php, java script, Node js,express**

2.6 Justification for selection of technologies

HTML is simple and best chosen for website development. It is used along with CSS and JavaScript. While there are other technologies, it still predominates all of them for creating web pages. Every browser supports HTML language. HTML can be integrated with many languages like PHP, JavaScript and Node.js

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Bootstrap is easy to set up and master, it has a lot of components, a good grid system, styling for many HTML elements ranging from typography to buttons, as well as support of JavaScript plugins, making it even more flexible.

MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company.

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

SQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance. MySQL is robust, secure and is easily available. MySQL is about flexibility and choice. Users have the ability to run MySQL on all major platforms, and to write applications in all popular language.

Chapter 3

Requirements and Analysis

3.1 PROBLEM DEFINATION

In the modern day of technology, most of the things can be automated or simplified using modern day solutions. This can be applicable in various fields. In our project we can have various task can be automated things like conducting admissions, generating marksheet, etc. Task related to conduction of exams can be simplifid, etc. The problem is to provide the complete information about the college campus. In which the college staff members, students and parents can access the information and will be familiar with college campus. It will provide interactive environment for the staff, students and parents by getting knowledge of student attendance, remarks, exams performances, grades, timetables, notices etc.

3.1.1.Time

A lot of time can be saved by using this college magement system as
Most of the things can automated and makes other task easier to accomplish. Various other tasks can be performed during that saved time

3.1.2.Manpower

The manpower required to complete tasks such as admissions in educational institute was much. But due to automation of various task
A lot of man power can be saved as the manual labour is reduced. And this may help the organization by efficiently managing the most precious resource in a business

3.1.3.Easy administration

The admistration of the institute can be easily done because evey thing can be monitored any time any where using any device supporting browser.
As the administration has a real time upto date information about the institute
Then the administrator can keep a follow ups.

3.2 Requirements Specification

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification consists of functional and non-functional requirements. It also contains use cases that describe user interactions that the software must provide. The software requirements specification document lists sufficient and necessary requirements for the project development. For this the developer needs to have clear and thorough understanding of the software. This is achieved through detailed and continuous communication with the project team and customer and analysing each and every activity of project in detail.

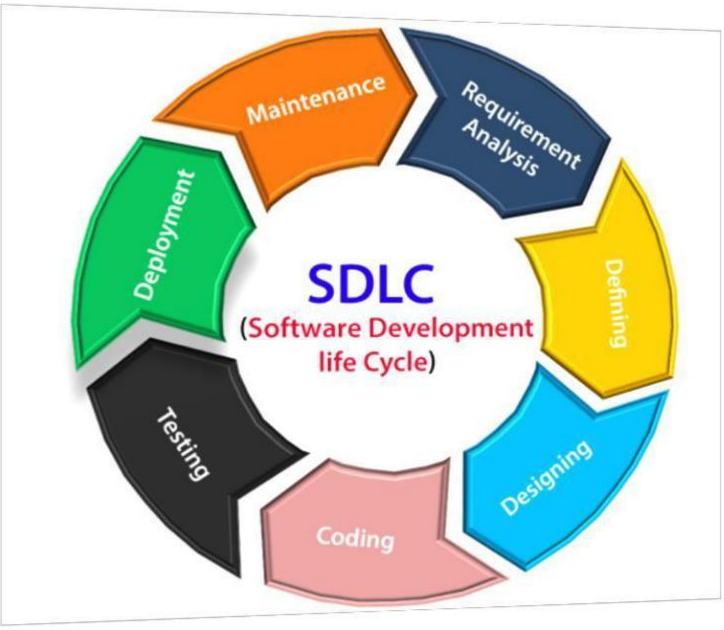
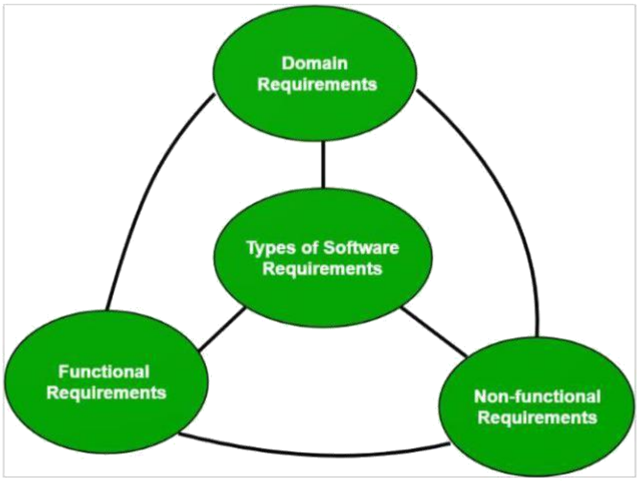


Figure – SDLC

Requirements specification or system analysis is part of Software Development life cycle in which we need to determine how the current information system functions and assess what users like to see in a new system. It has two sub-phases, viz. requirements determination and requirements structuring.



3.2.1FUNCTIONAL REQUIREMENTS

In software and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs. It specifies particular results of a system. These are in contrast with non-functional requirements, which specify overall characteristics.

Functional requirements describe how a product must behave, what its features and functions. Since this project uses database & control, it needs the retrieval of information from the database.

Academic activities include:

1. Academic Programme : Applications and Admissions, Enrolling new students, Course registration (enrollment for desired courses) and Fee payment, Evaluation and Grading, Viewing and printing grade cards.
2. Student Information : Student records management.

Administrative activities consist of:

1. Employee Information : Maintaining Employee records, Accessing data and Generating reports as required.
2. Human Resources Management : Faculty and Staff Recruitment Process

3. General Administration : File movement Tracking (across departments/administration), Facility to raise an issue / ticket and track it, Meeting Management, Managing Institute Advertisements.

Accounting and Finance includes:

1. Student account management covering fees, teaching assistantship, scholarships, fines and other charges
2. Employee Accounts Management: Salaries, Leave management, Income Tax management, Pension management, TA/DA and PDA.

3.2.2NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe the general characteristics of a system. They are also known as quality attributes.

In software and systems engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are often called as quality attributes‘. They are contrasted with functional requirements that define specific behaviour or functions. Following are the non-functional requirements of our project:

1. Security

   Nobody should be allowed to tamper with data; Enhanced Security for sensitive data. It should be made sure that only users who are given specific rights can access data and all actions are logged, thus providing an extensive role based authorization. Note: Password security may not be sufficient. Advanced technologies like use of biometrics is desirable.
2. Backup

   There should be an easy back-up feature for the entire data, to prevent losing any data. 4. Platform/Browser independence The system should be able to work on any of the modern browsers like Firefox / Explorer / Opera / Chrome, and any of the common Operating Systems like Linux, Windows and Mac OS. 3

3. Data migration

   (i) There should be an easy way to migrate data from the current system to a new system.
   (ii) The system should be able to interface with Microsoft Excel – read from Excel and write to Excel. Standard file formats such as CSV or XML should also be supported for both import and export. 6.

4. Performance

   During Course Registration an estimate of 10,000 students should be able to register over an interval of 2 days. (ii) The system should handle the simultaneous usage of almost 1000 users at a time.

3.4 <u>Software and hardware requirements</u>

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements. Most software defines two sets of system requirements: minimum and recommended.

3.4.1 <u>Software requirements</u>

A software requirement is a field within software engineering that deals with establishing the needs of stakeholders that are to be solved by software.

Programming Language: Php, java script, Node.js, express

Web Technology: Html/CSS , bootstrap, react js

Operating System: Windows XP, Windows 7(ultimate, enterprise).

Database: SQL Server 2012

Browser Support:

IE 9+

Firefox (latest)

Chrome (latest)

3.4.2 <u>Hardware Specification:</u>

Processor – i3.

Hard Disk – 5 GB.

Memory – 1GB RAM.

Processor Speed- Dual Core 2GHz

Note:(Hardware and software requirements mentioned here are required to develop the project efficiently, where as Browser that support this project i.e. on which the project can be used by users are mentioned in Browser Support part above)

Gantt Chart

In the Gantt chart we show the time spent for each phase of the software development. Gantt charts are a project-planning tool that can be used to represent the timing of tasks required to complete a project. Because Gantt charts are simple to understand and easy to construct, they are used by most project managers for all but the most complex projects.

In a Gantt chart, each task takes up one row. Dates run along the top in increments of days, weeks or months, depending on the total length of the project. The expected time for each task is represented by a horizontal bar whose left end marks the expected beginning of the task and whose right end marks the expected completion date. Tasks may run sequentially, parallel or overlapping. As the project progresses, the chart is updated by filling in the bars to a length proportional to the fraction of work that has been accomplished on the task. This way, one can get a quick reading of project progress by drawing a vertical line through the chart at current date. Completed tasks lie to the left of the line and are completely filled in. Current tasks cross the line and are behind schedule if their filled-in section is to the left of the line and ahead of schedule if the filled-in section stops to the right of the line.

Future tasks lies completely to the right of the line. In constructing a Gantt chart, keep the tasks to a manageable number (no more than 15 or 20) so that the chart fits on a single page

.More complex projects may require subordinate charts which detail the timing of all the subtasks which make up one of the main tasks.

For team projects, it often helps to have an additional column containing numbers or initials, which identify that on the team is responsible for the task. Often the project has important events, which you would like to appear on the project timeline, but which are not tasks. For example, you may wish to highlight when a prototype is complete or the date of a design review. We planned our project using according to the Gantt chart as shown as follows:

Task Name	Start Date	End Date	Duration	Resources
Project Topic Selection	07 July 2021	16 July 2021	10	Ankit, Rohan
Feasibility Study	17 July 2021	23 July 2021	7	Rohan
Requirement Gathering	24 July 2021	03 August 2021	10	Ankit
Survey Of Technology	04 August 2021	08 August 2021	5	Rohan
Analysis	09 August 2021	23 August 2021	15	Ankit
Planning	24 August 2021	30 August 2021	7	Ankit, Rohan
Architectural Design	31 August 2021	06 September 2021	7	Ankit, Rohan
Detail Design	07 September 2021	26 September 2021	20	Ankit, Rohan
Coding	22 November 2021	20 January 2022	60	Ankit, Rohan
Testing	21 January 2022	09 February 2022	20	Ankit, Rohan
Finalization	10 February 2022	20 February 2022	10	Ankit, Rohan

Figure- Gantt Chart Table

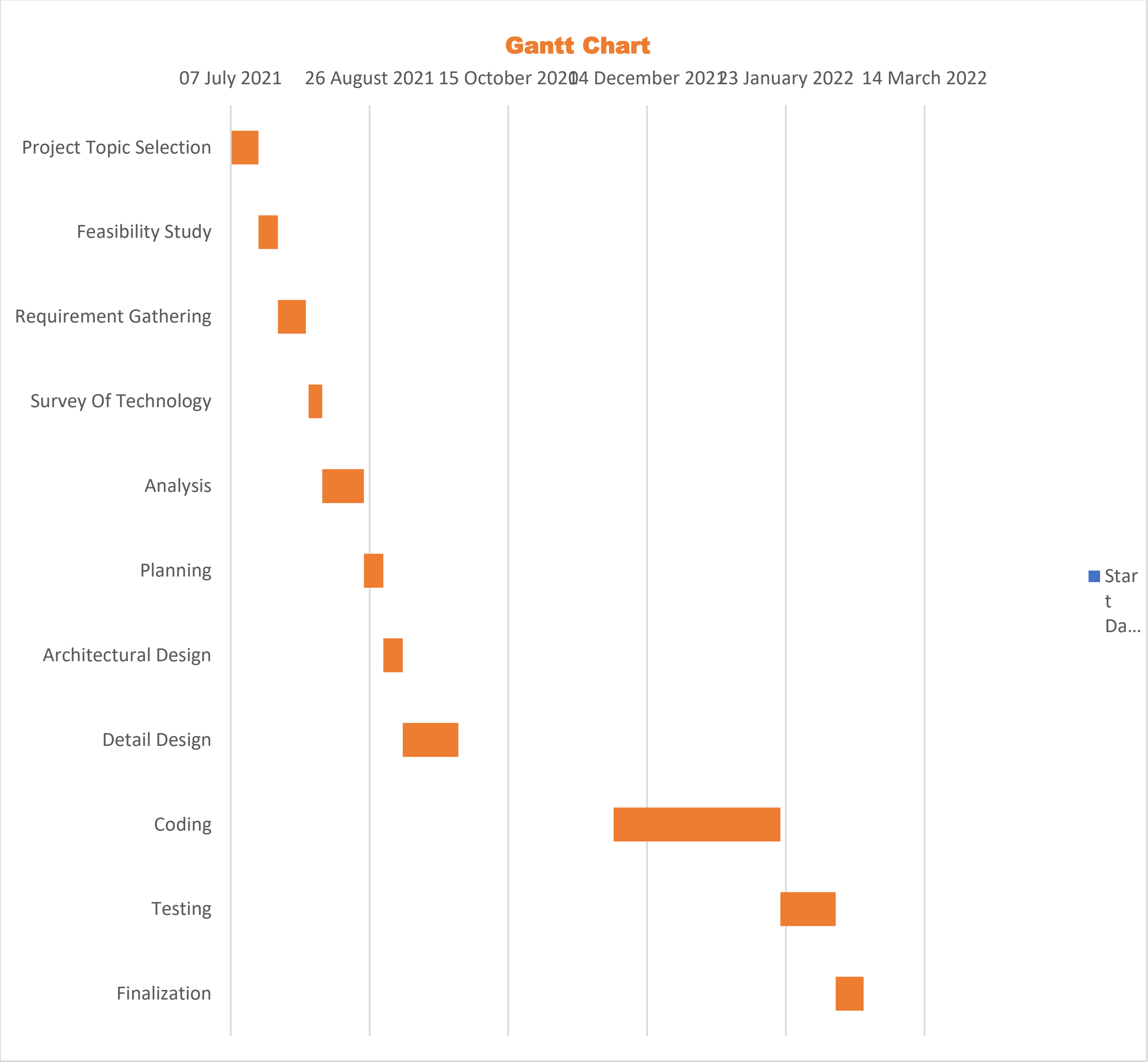


Figure- Gantt Chart

Program Evaluation Review Technique (PERT)

The program (or project) evaluation and review technique (PERT) is a statistical tool used in project management, which was designed to analyze and represent the tasks involved in completing a given project. Figure shows PERT diagram for activities. PERT diagram shows activity no, target date, expected date and standard deviation of each activity. The activities involved are search for project, project approval, planning, requirement & analysis, documentation, designing, designing complete, database designing, documentation complete, coding & implementation and last testing.

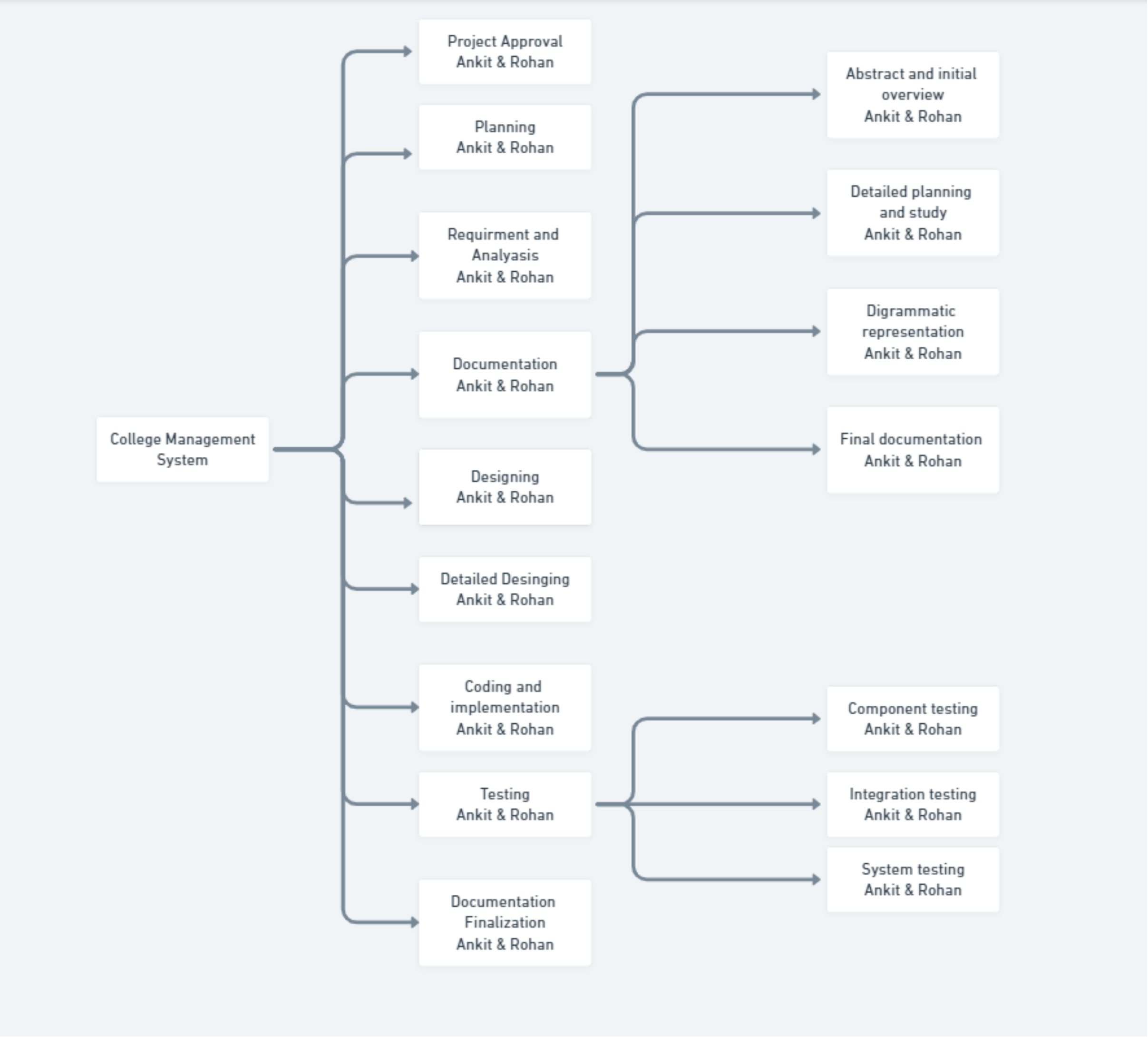

College Management System
Project Approval Ankit & Rohan
Planning Ankit & Rohan
Requirment and Analyasis Ankit & Rohan
Documentation Ankit & Rohan
Designing Ankit & Rohan
Detailed Desinging Ankit & Rohan
Coding and implementation Ankit & Rohan
Testing Ankit & Rohan
Documentation Finalization Ankit & Rohan
Abstract and initial overview Ankit & Rohan
Detailed planning and study Ankit & Rohan
Digrammatic representation Ankit & Rohan
Final documentation Ankit & Rohan
Component testing Ankit & Rohan
Integration testing Ankit & Rohan
System testing Ankit & Rohan

Figure- Pert Char

3.5 Preliminary product description

- **Reliability**

This system is designed to have very simple database just to cater the exact need of online hair salon system. It is tested for all the constraints at development stage.

- **Security**

This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use.

3.6 Conceptual models

A conceptual model is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents.

Entity relationship diagram

The entity – relationship (ER) data model allows us to describe the data involved in a real world enterprise in terms of object and their relationships and is widely used to develop an initial database design.

E-R diagram is a relationship between two entity sets. E-R diagram can express the overall structure of a database graphically. E-R diagrams are simple and clear.

The ER model is important primarily for its role in database design. It provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed and precise description that can be implemented in a DBMS. The ER model is used in a phase called ―Conceptual Database Design‖. It should be noted that many variations of ER diagrams are in use and no widely accepted standards prevail. ER modelling is something regarded as a complete approach to design a logical database schema.

This is incorrect because the ER diagram is just an approximate description of data, constructed through a very subjective evaluation of the information collected during requirements analysis.

Entity:

An entity is an object in the real world that is distinguishable from other objects. Examples include the following: The C++ language, the address of the manager of the institution. It is often useful to identify a collection of similar entities. Such a collection is called as ―Entity set‖. Note that entity set need not be disjoint.

An entity is a ―thing‖ or ―object‖ in the real world that is distinguishable from all set of objects. An entity set is a set of the same type that share same properties, or attributes. An entity is represented by a set of attributes.

Attributes:

An entity is described using a set of attributes. All entities in a given entity set have the same attributes; this essentially what we mean by similar. Our choice of attributed reflects the level of detail at which we wish to represent information in crisis.

Domain:

For each attribute associated with an entity set, we must identify a domain of possible values.

Key:

Further, for each entity set we choose a key. A key is a minimal set of attributed whose values uniquely identify an entity in the set. There could be more than one candidate; if so we designate one of them as primary key.

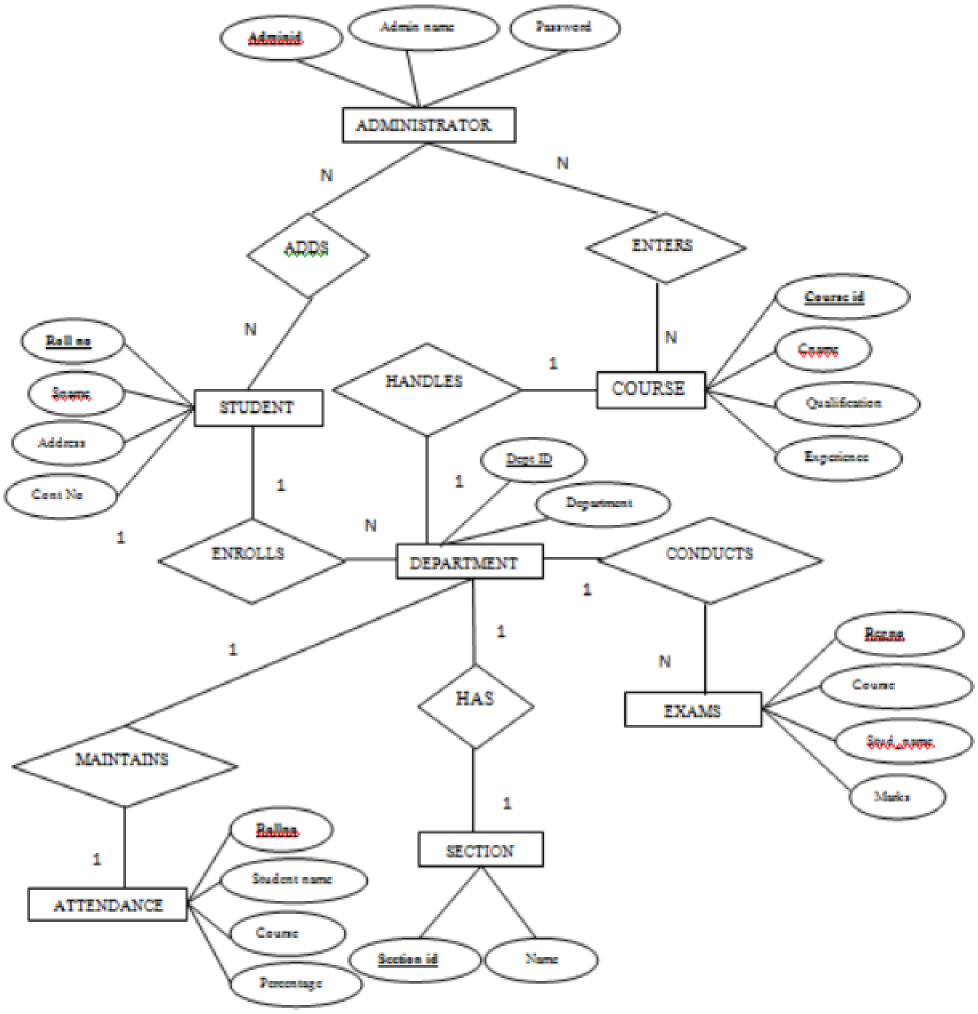


Figure- ER Diagram1

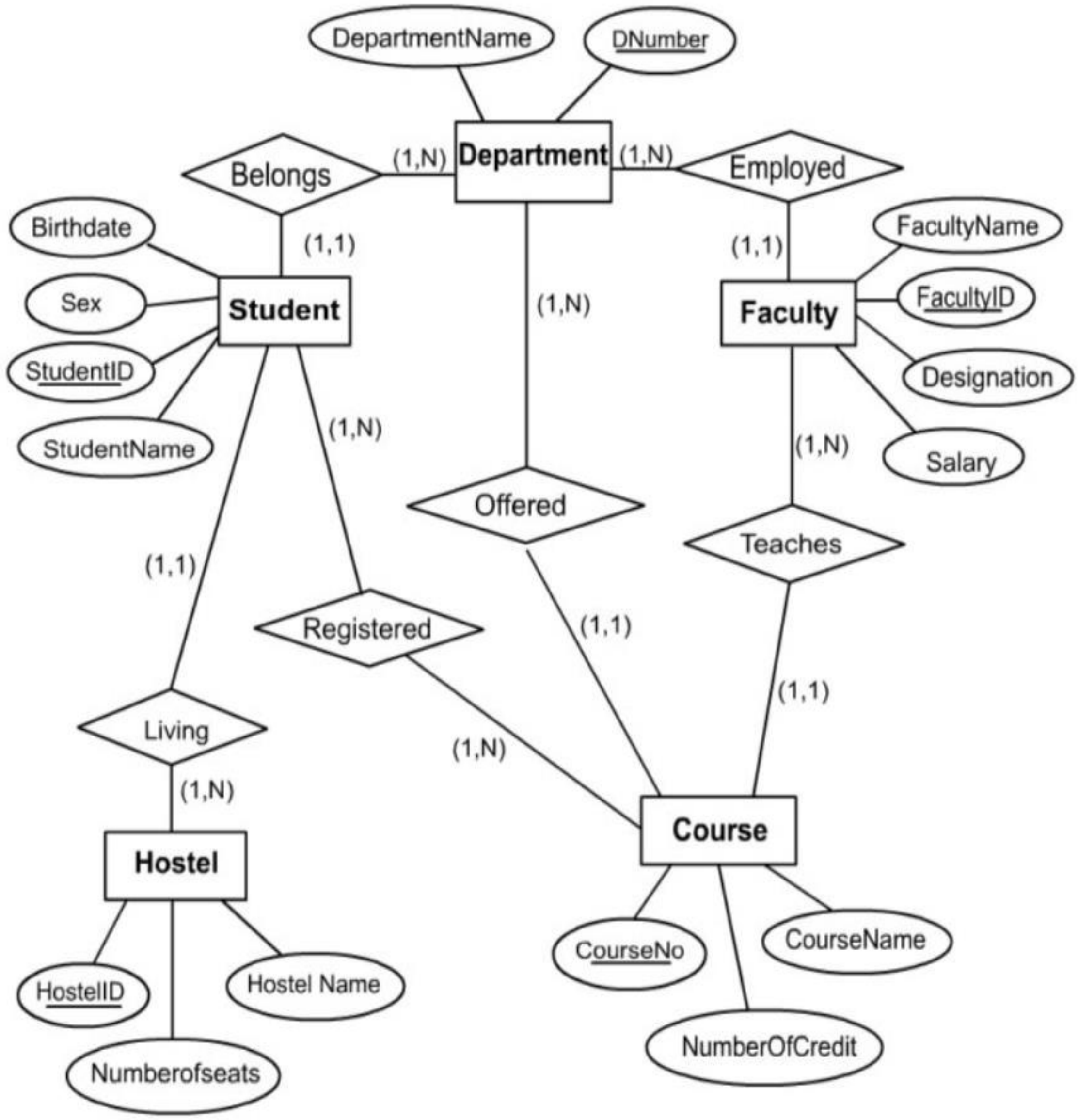


Figure- ER Diagram2

Use case

Use Case Diagram is an expression of relations between the use cases in a specific system or object and the external actors. Use Case expresses the functions of the system and how the system functions interact with the external actors.

Actor represents the one who is interacting with the system and use-cases. Use case represents system functions.

The 2 main components of use case diagram are use case and actor.

PURPOSE:

1) Specify the context of the system.

2) Capture the requirements of system.

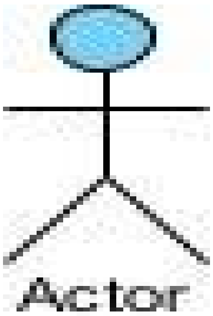
3) Validate the system architecture.

4)Drive implementation & generate test cases.

5)Developed analyst & domain expert.

ELEMENTS:

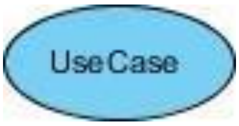
Actor



- Someone interacts with use case (system function).
- Named by noun.
- Actor triggers use case(s).

- Actor has a responsibility toward the system (inputs), and Actor has expectations from the system (outputs).

Use Case



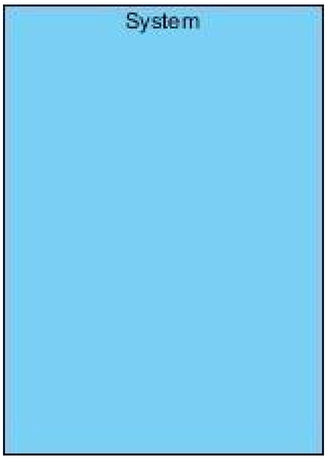
- System function (process - automated or manual)
- Named by verb + Noun (or Noun Phrase).
- i.e. Do something
- Each Actor must be linked to a use case, while some use cases may not be linked to actors.

Communication Link

- The participation of an actor in a use case is shown by connecting an actor to a use case by a solid link.
- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Boundary of system

- The system boundary is potentially the entire system as defined in the requirements document.
- For large and complex systems, each module may be the system boundary.

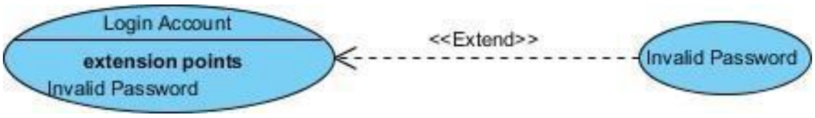


- For example, for an ERP system for an organization, each of the modules such as personnel, payroll, accounting, etc.

- Can form a system boundary for use cases specific to each of these business functions.

The entire system can span all of these modules depicting the overall system boundary.

Extends

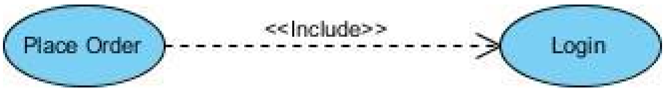


- Indicates that an **"Invalid Password"** use case may include (subject to specified in the extension) the behaviour specified by base use case **"Login Account"**.

- Depict with a directed arrow having a dotted line. The tip of arrowhead points to the base use case and the child use case is connected at the base of the arrow.

- The stereotype "<<extends>>" identifies as an extend relationship

Include



- When a use case is depicted as using the functionality of another use case, the relationship between the use cases is nam ed as include or uses relationship.

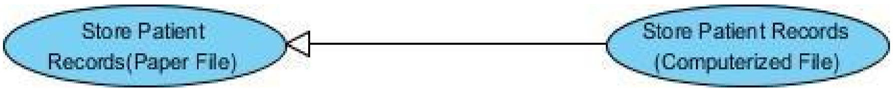
- A use case includes the functionality described in another use case as a part of its business process flow.

- A uses relationship from base use case to child use case indicates that an instance of the base use case will include the behaviour as specified in the child use case.

- An include relationship is depicted with a directed arrow having a dotted line. The tip of arrowhead points to the child use case and the parent use case connected at the base of the arrow.

- The stereotype "<<include>>" identifies the relationship as an include relationship.

Generalization


Store Patient Records(Paper File)
Store Patient Records (Computerized File)

- A generalization relationship is a parent-child relationship between use cases.

- The child use case is an enhancement of the parent use case.

- Generalization is shown as a directed arrow with a triangle arrowhead.

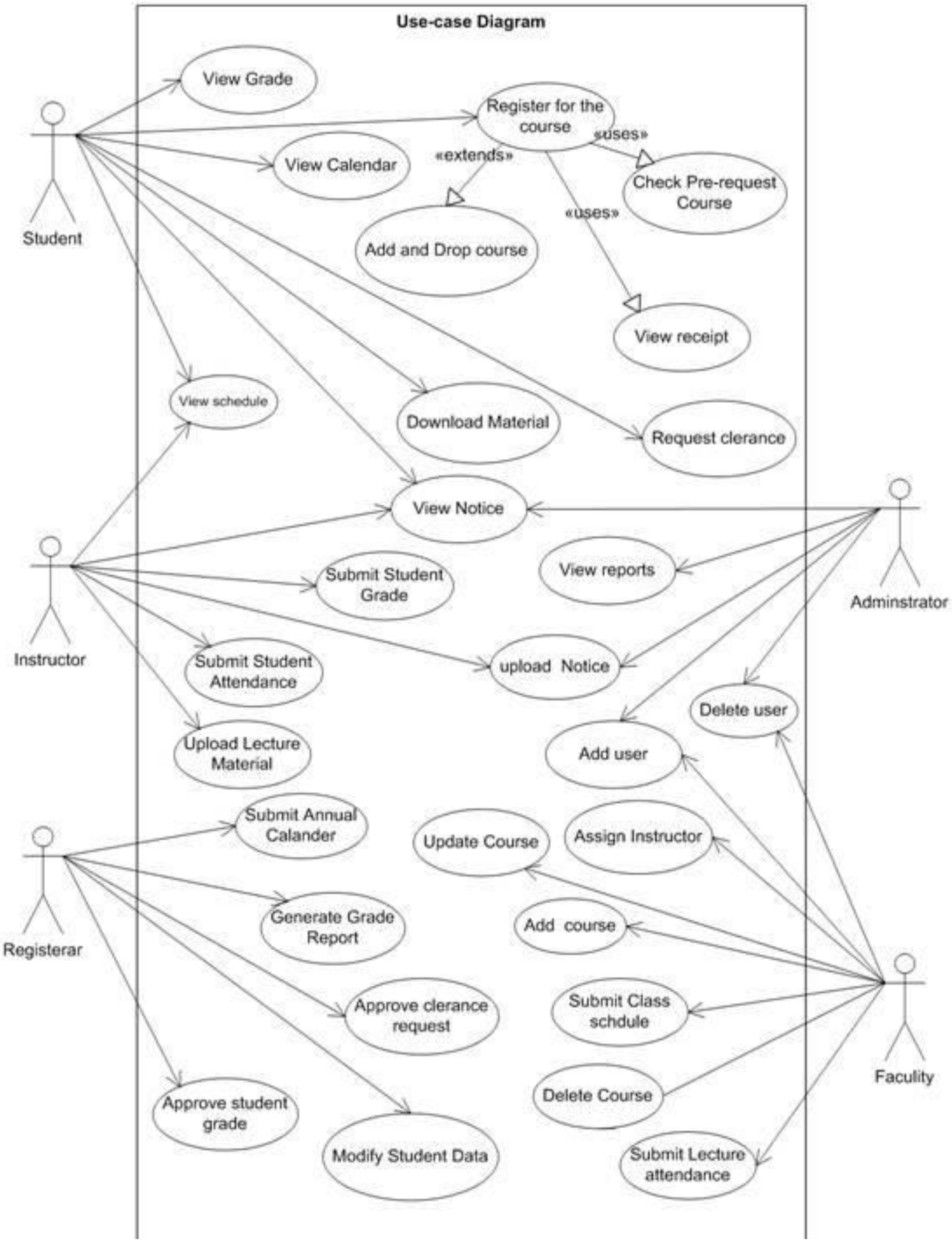


Figure- Use case Diagram

Sequence Diagram

Sequence diagrams describe how objects, or groups of objects, interact within a system. Interacting objects can, for example, be classes, program components or real world instances such a customer who is buying a train ticket in the station.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. Sequence diagrams are used as part of the analysis and design process. The Together application has two primary modes:

Sequence Diagram Generation

Diagrams as ―sketches‖ Diagrams with objects and messages associated with classes and operations, respectively. Developers can use sequence diagrams to document the database dynamic aspects of an object model. In the following examples, sequence diagrams are used to make object interactions more understandable by tackling the dynamics behind major business methods. This technique frequently leads to improvements in the object model. For example, it can reveal missing associations to objects that you need to message, missing methods in a class, and sometimes even missing classes.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence Diagram Notations –

1. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

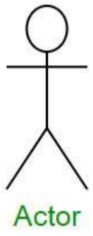


Figure – notation symbol for actor

We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram. For example – Here the user in seat reservation system is shown as an actor where it exists outside the system and is not a part of the system.

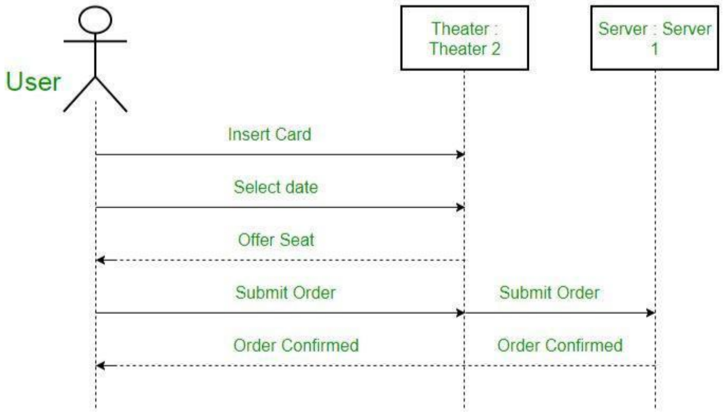


Figure – an actor interacting with a seat reservation system

2. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name : Class Name

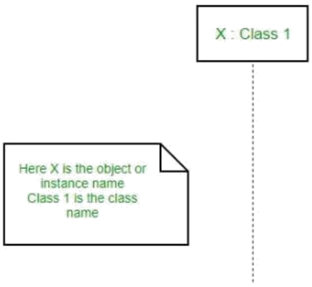

X : Class 1
Here X is the object or instance name Class 1 is the class name

Figure – lifeline

We display a lifeline in a rectangle called head with its name and type. The head is located on top of a vertical dashed line (referred to as the stem) as shown above. If we want to model an unnamed instance, we follow the same pattern except now the portion of lifeline's name is left blank.

Difference between a lifeline and an actor – A lifeline always portrays an object internal to the system whereas actors are used to depict objects external to the system. The following is an example of a sequence diagram:

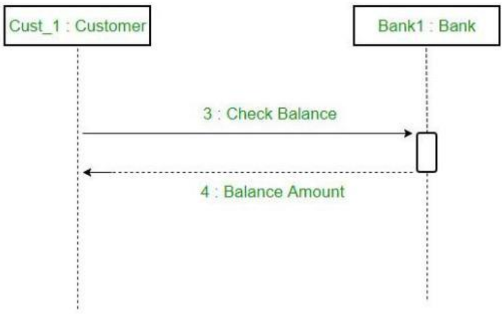
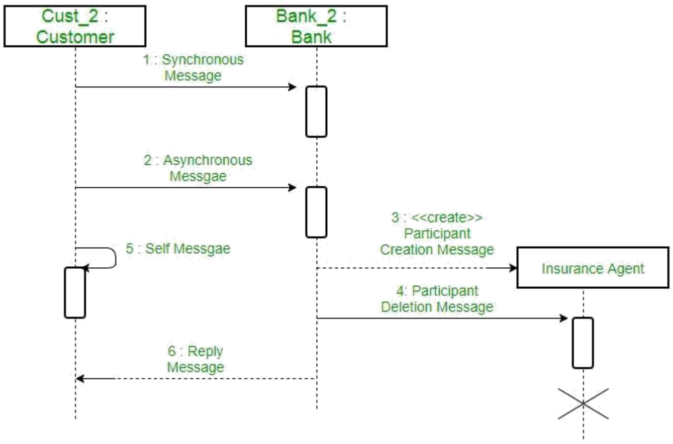


Figure – a sequence diagram

3. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram. Messages can be broadly classified into the following **categories** :



Uses of sequence diagrams –

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

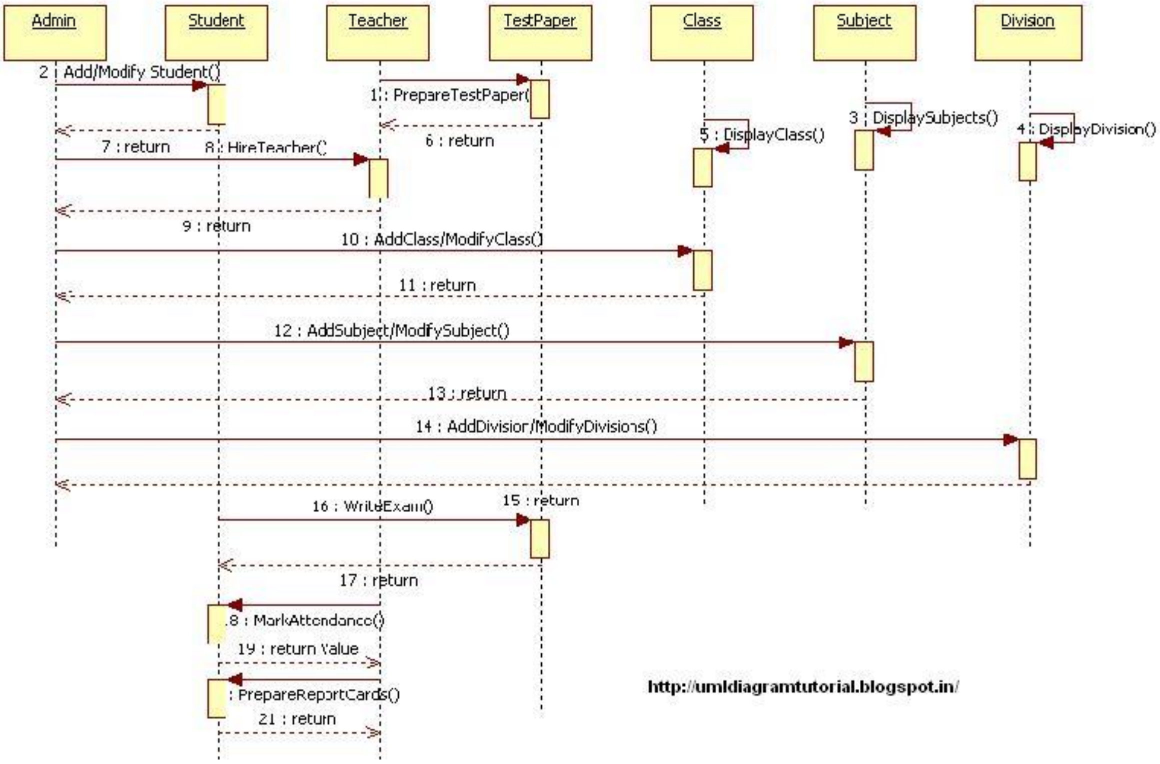


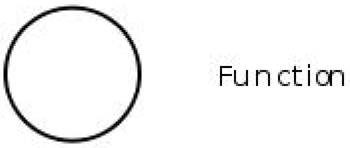
Figure- Sequence Diagram

Data flow diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).Diagram notations include entity, process, data store and data flow. Entity is human or system involved in the process. Process is the task which the diagram represents. Data source is data required by process. Data flow acts as connectors which connects entities, processes and data sources together. Figure shows DFD diagram for free lancing system. It has two levels: context level and level 1. Context level is simple showing two entities and one process connected through data flow. DFD components

DFD consists of processes, flows, warehouses, and terminators. There are several ways to view these DFD components.

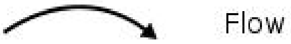
Process



The process (function, transformation) is part of a system that transforms inputs to outputs. The symbol of a process is a circle, an oval, a rectangle or a rectangle with rounded corners (according to the type of notation). The process is named in one word, a short sentence, or a phrase that is clearly to express its essence.

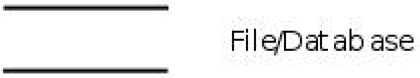
Data Flow

Data flow (flow, dataflow) shows the transfer of information (sometimes also material) from one part of the



system to another. The symbol of the flow is the arrow. The flow should have a name that determines what information (or what material) is being moved. Exceptions are flows where it is clear what information is transferred through the entities that are linked to these flows. Material shifts are modelled in systems that are not merely informative. Flow should only transmit one type of information (material). The arrow shows the flow direction (it can also be bi-directional if the information to/from the entity is logically dependent - e.g. question and answer). Flows link processes, warehouses and terminators.

Warehouse



The warehouse (data store, data store, file, database) is used to store data for later use. The symbol of the store is two horizontal lines, the other way of view is shown in the DFD Notation.

The name of the warehouse is a plural noun (e.g. orders) - it derives from the input and output streams of the warehouse. The warehouse does not have to be just a data file, for example, a folder with documents, a filing cabinet, and optical discs.

Terminator

The Terminator is an external entity that communicates with the system and stands outside of the system. It can be, for example, various organizations (e.g. a bank), groups of people (e.g. customers), authorities (e.g. a tax office) or a department (e.g. a human-resources department) of the same organization, which does not belong to the model system. The terminator may be another system with which the modelled system communicates.

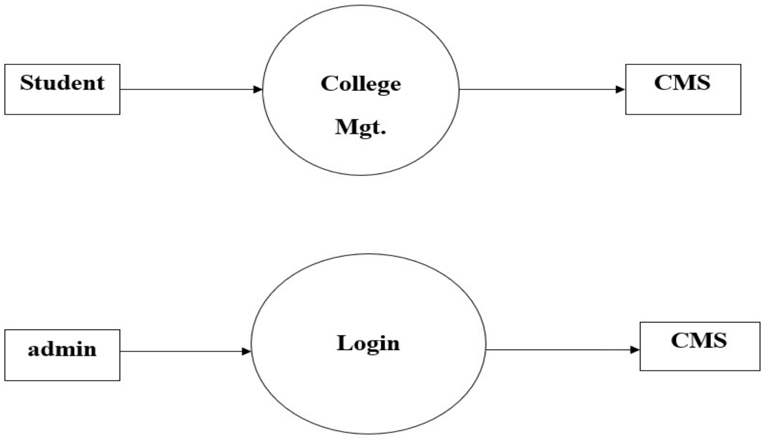


Figure- 0 level DFD

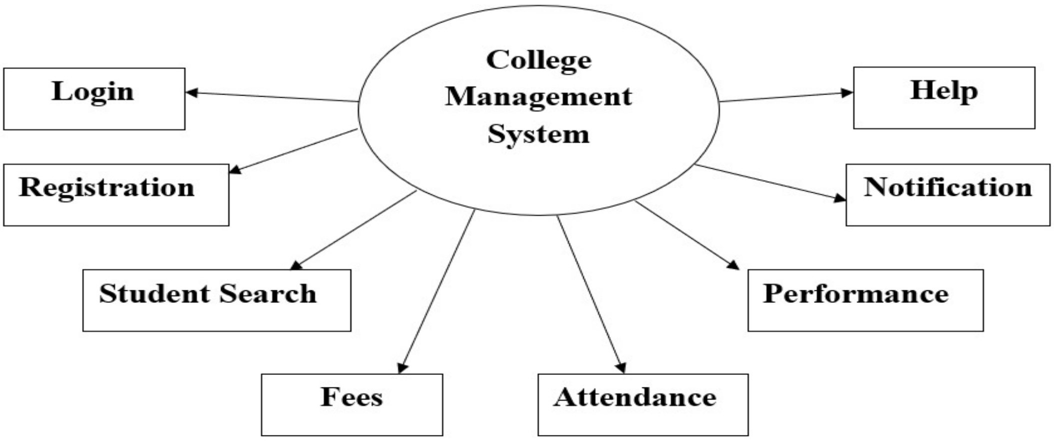


Figure- 1 level DFD

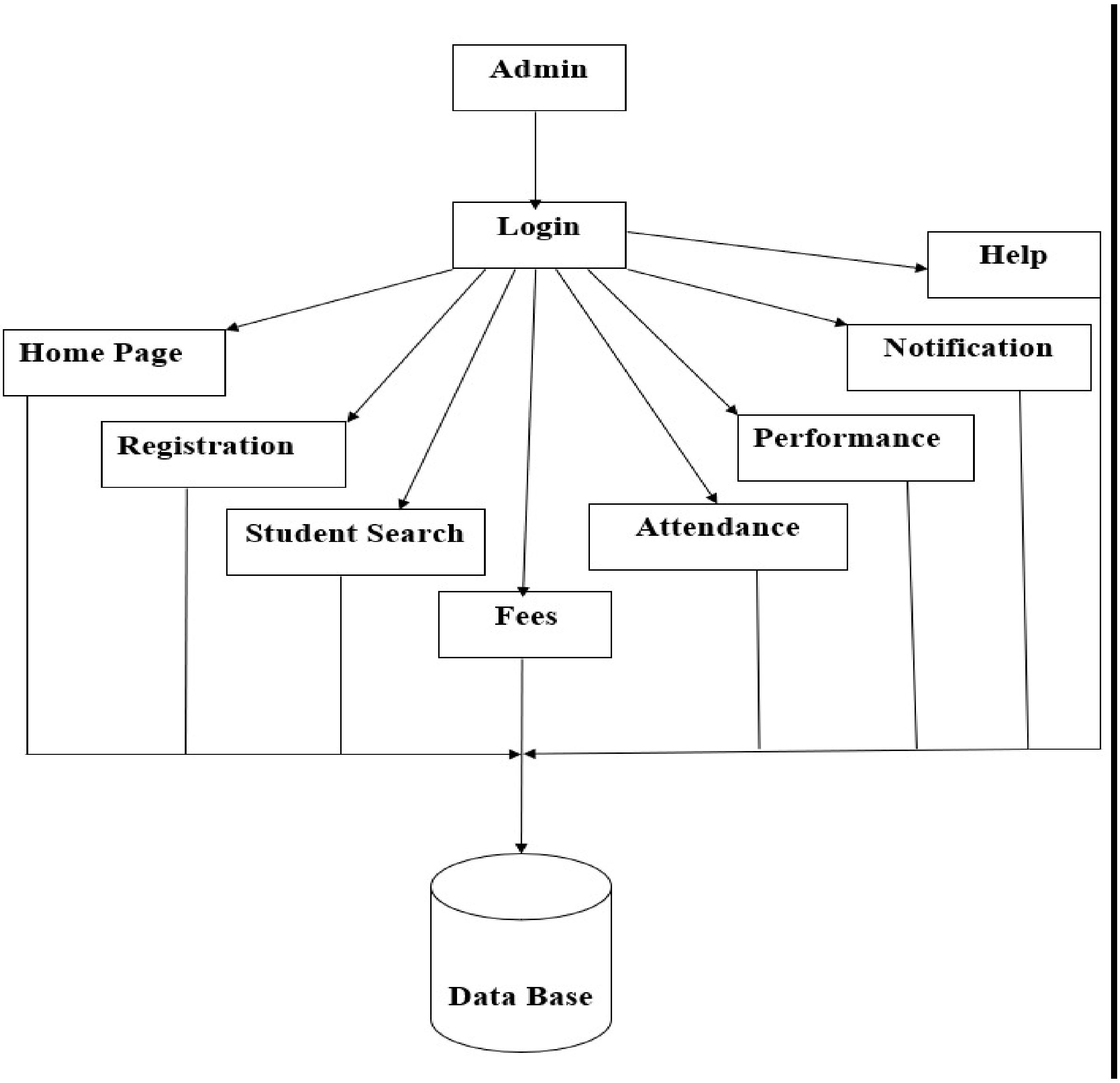


Figure- 2 level DFD

System flowcharts

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

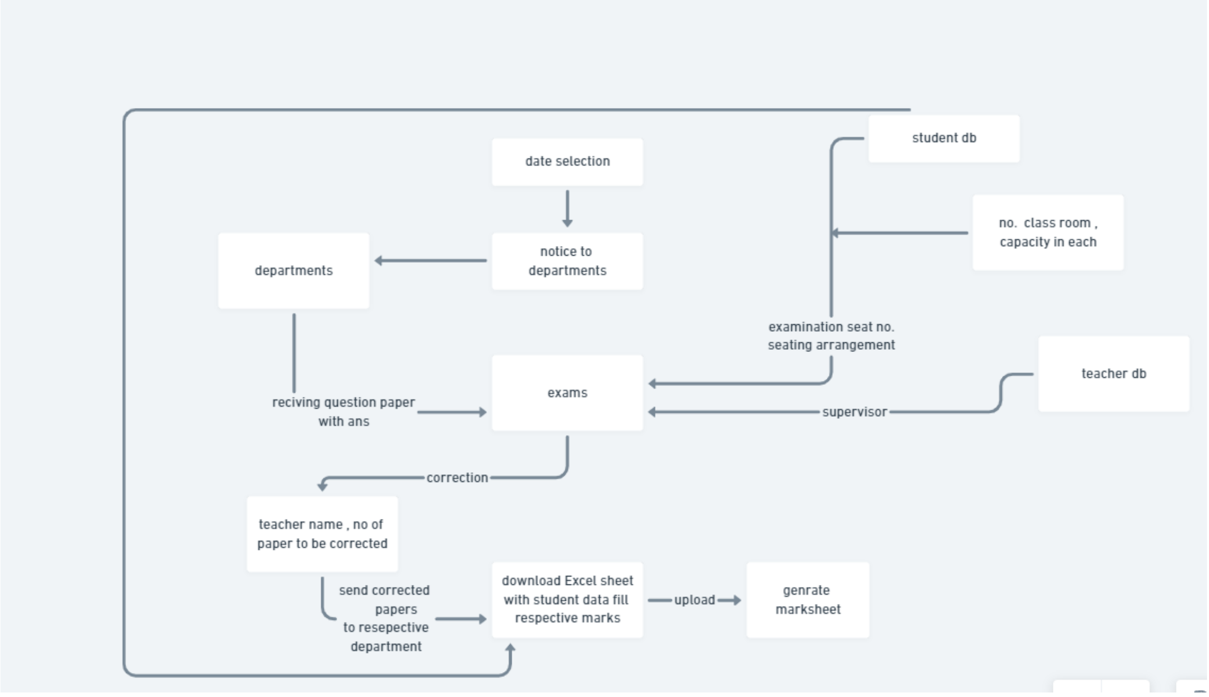

student db
date selection
no. class room , capacity in each
departments
notice to departments
examination seat no. seating arrangement
teacher db
exams
reciving question paper with ans
supervisor
correction
teacher name , no of paper to be corrected
send corrected papers to resepective department
download Excel sheet with student data fill respective marks
upload
genrate marksheet

Figure- System Flowchart

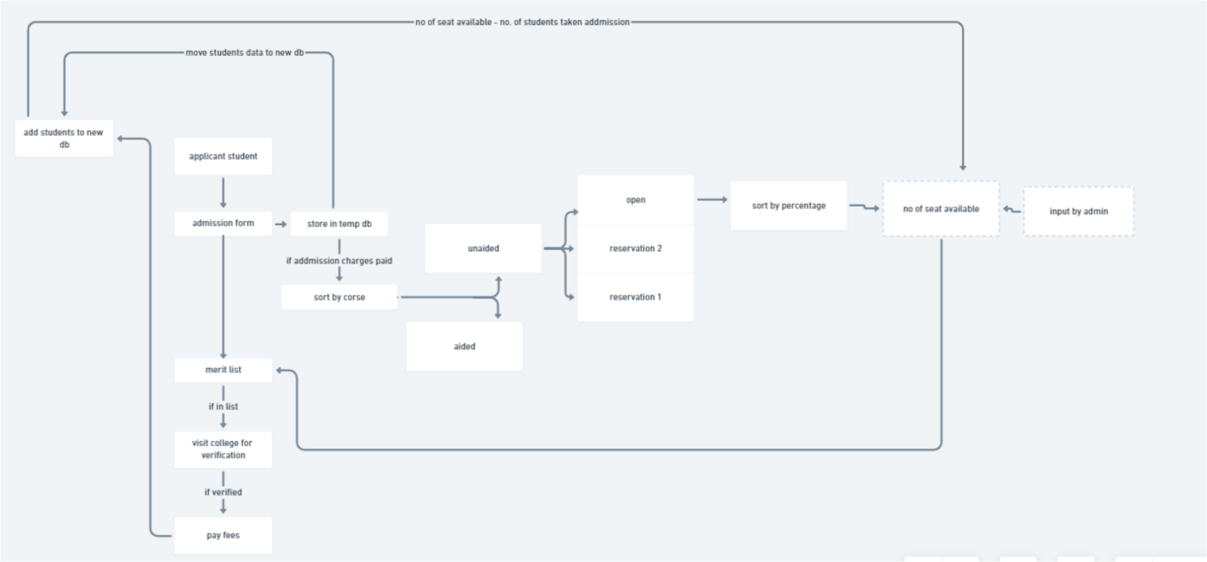


Figure- System Flowchart

Class diagram

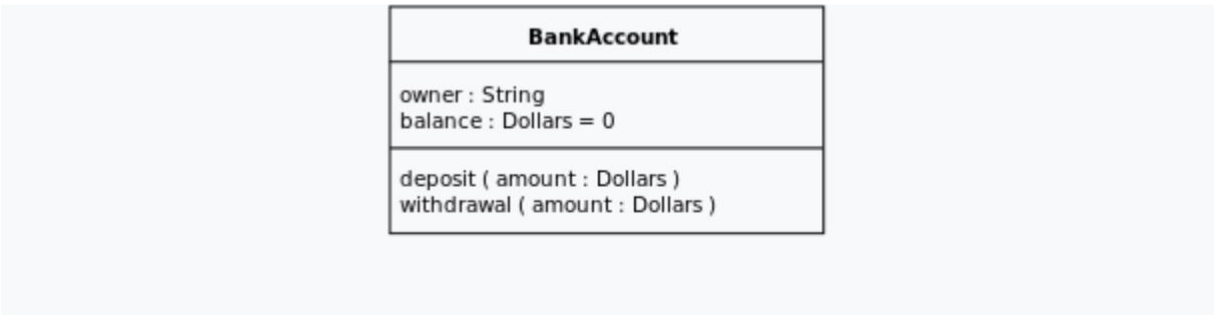
Class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments: • The top compartment contains the name of the class. It is printed in bold and centred, and the first letter is capitalized.

- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.



A class with three compartments.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

In order to further describe the behaviour of systems, these class diagrams can be complemented by a state diagram or UML state machine.

Purpose of Class Diagrams: -

1. Shows static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective

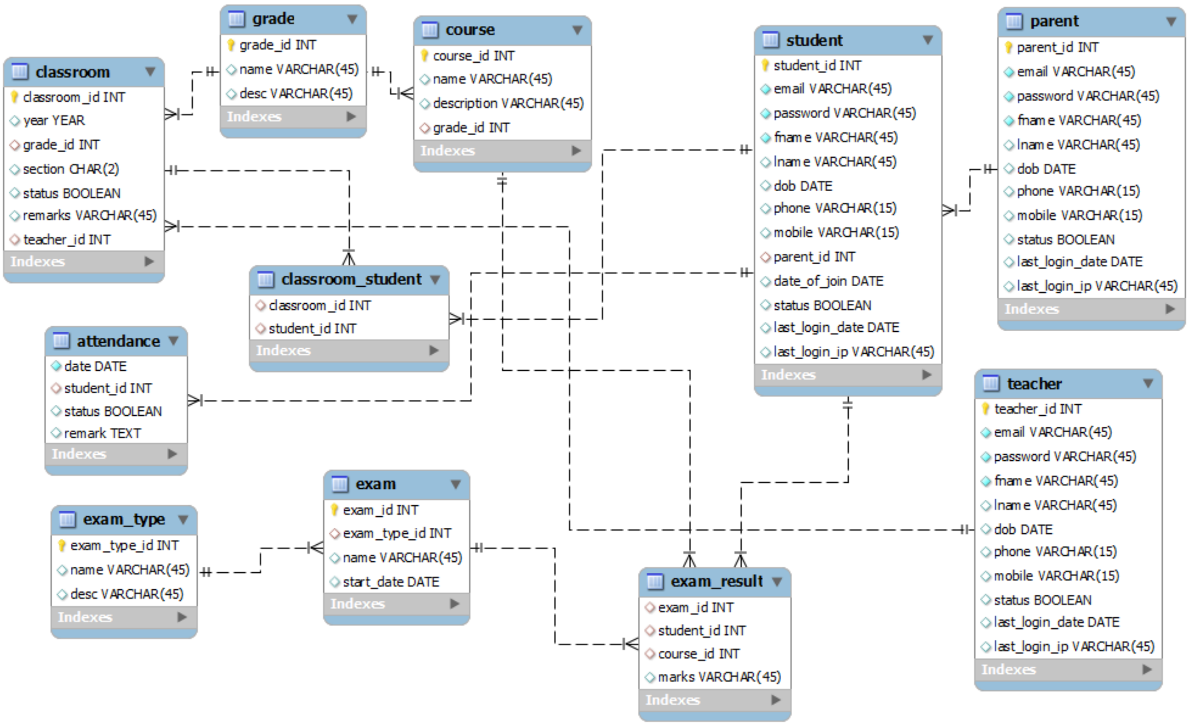


Figure- Class Diagram

Chapter 4

SYSTEM DESIGN

4.1 Basic module

What is module?

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. An enterprise-level software application may contain several different modules, and each module serves unique and separate business operations.

Modules make a programmer's job easy by allowing the programmer to focus on only one area of the functionality of the software application. Modules are typically incorporated into the program (software) through interfaces

Modular Features

ONLINE REGISTRATION & ADMISSIONS

- Registered Student List
- Merit list- General, Category-wise, and Quota
- Online Admission Register
- Current Admission Position
- Admission Cancellation Report
- MIS Reports According to Fee Type, Medium, Gender, Caste, Category, Religion, Region

ONLINE FEE PAYMENT

- e-Receipts of Fee Payments
- Fee Collection Reports
- Outstanding Fee Reports

STUDENT ADMINISTRATION

- Identity Card & Certificates like TC, Bonafide, NOC, DOB, Attempt
- College Data Export to Excel
- MIS Reports According to Faculty, Medium, Gender, Caste, Category
- Student Lists Management for Colleges
- Subject/Class-wise Attendance Sheet
- Document Submission Report

TIMETABLE & STUDENT ATTENDANCE

- Timetable According to Course, Teacher & Classroom
- Attendance Reports: According to Subject, Faculty, Class
- Poor Attendance Report & Undertaking Report
- Attendance Analysis According to Subject, Faculty, and Class
- SMS/Email to Parents/Students

PAYROLL MANAGEMENT

- Pay Slips & Salary Certificates
- Salary Register & Abstract
- Schedules of Deductions-PT, LIC
- Annual Salary Reports
- MIS Reports Generation

ATTENDANCE & LEAVE

- Leave & Attendance Reports
- Date-wise, Month-wise Leave & Attendance Reports
- Daily Absentee Reports
- Continuous Absentee Reports
- Department-wise Leave & Attendance Reports

EARNING MANAGEMENT SYSTEM (LMS)/ E-LEARNING

Share Information & Send Announcements

Create Question Banks & e-Library

E-content & Teaching Notes Upload Provision

Assign & Evaluate Homework & Other Tasks Online

4.2 **Data design**

What is database?

A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it.

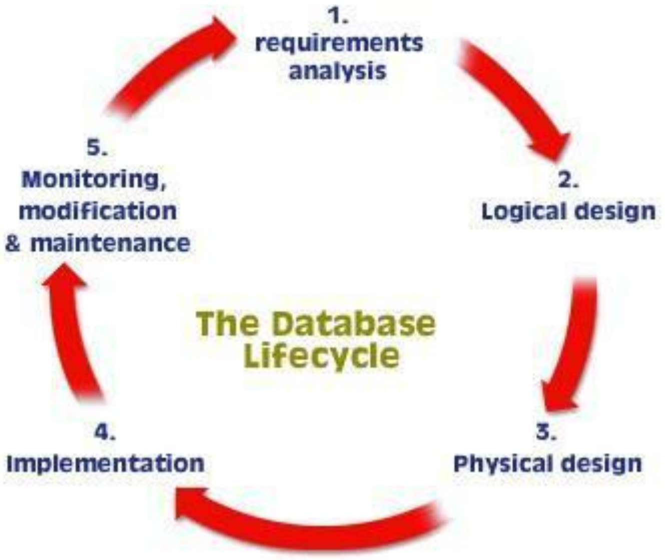


Figure-Database Development Life Cycle

Database Development Life Cycle

As stated above, a database system is a fundamental component of the larger enterprise information system. The database development life cycle (DDLC) is a process of designing, implementing and maintaining a database system to meet strategic or operational information needs of an organization or enterprise such as:

- Improved customer support and customer satisfaction.
- Better production management.
- Better inventory management.
- More accurate sales forecasting.

The database development life cycle (DDLC) is inherently associated with the software development life cycle (SDLC) of the information system. DDLC goes hand-in-hand with the SDLC and database development activities starts right at the requirement phase

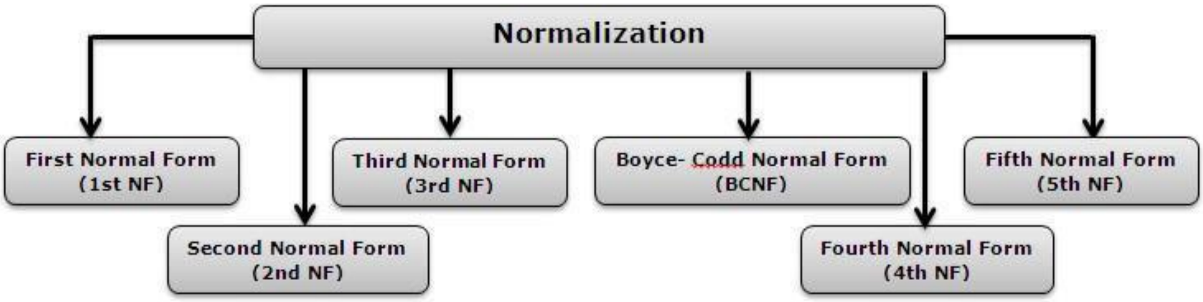
What is a Database Management System (DBMS)?

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data. A DBMS relieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database.

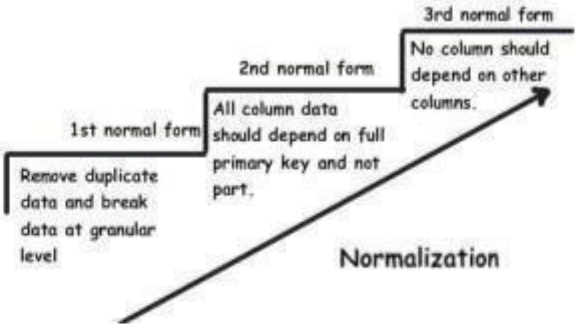
Description of Normalization

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

Types of Normalization


Normalization
First Normal Form (1st NF)
Second Normal Form (2nd NF)
Third Normal Form (3rd NF)
Boyce- Codd Normal Form (BCNF)
Fourth Normal Form (4th NF)
Fifth Normal Form (5th NF)

We need to know three stages of normalization; a basic outline of the three stages is shown in the image below. Normalization is used to keep a database as efficient as it can possibly be without any unnecessary duplication and no redundant data. When working with large volumes of data this can cause a lot of space and time to be saved as its less data to process. 1NF-Atomic Data Test 2NF-Partial Dependency Test 3NF-NonKey Dependency Test


3rd normal form
No column should depend on other columns.
2nd normal form
All column data should depend on full primary key and not part.
1st normal form
Remove duplicate data and break data at granular level
Normalization

Advantages of Third Normal Form (3NF) Reduced data redundancy**,** increased data quality**,** capturing complete business requirements**,** data modification anomalies (insert, delete and update) is reduced.

4.2.1 Schema Design

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed.

Table Name: Student

Field	Description	Type	Length
stud_ID (PK)	Student ID	Int	11
fname	Student First Name	Varchar	255
lname	Student Last Name	Varchar	255
gender	Student Gender	Int	11
age	Student Age	Int	11
contact_add	Contact Address	Int	11
stud_email	Student Email	Varchar	255
stud_pass	Student Password	Varchar	255

Table Name: Instructor

Field	**Description**	**Type**	**Length**
ins_ID (PK)	Instructor ID	Int	11
fname	Instructor First Name	Varchar	255
lname	Instructor Last Name	Varchar	255
gender	Instructor Gender	Int	11
age	Instructor Age	Int	11
contact_add	Contact Address	Int	11
ins_email	Instructor Email	Varchar	255
ins_pass	Instructor Password	Varchar	255

Table Name: Course

Field	**Description**	**Type**	**Length**
course_ID (PK)	Course ID	Int	11
course_name	Course Name	Varchar	30
course_desc	Course Description	Varchar	30
school_yr	School Year	Int	11

Table Name: Subjects

Field	**Description**	**Type**	**Length**
sub_ID (PK)	Subject ID	Int	11
name	Subject Name	Varchar	255
course_ID (FK)	Course ID	Int	11

Table Name: Schedules

Field	**Description**	**Type**	**Length**
sched_ID (PK)	Schedule ID	Int	11
course_ID (FK)	Course ID	Int	11
sub_ID (FK)	Subject ID	Int	11
ins_ID (FK)	Instructor ID	Int	11
stud_ID (FK)	Student ID	Int	11
day	Day of schedule	Date	
time_start	Starting Time	Time	
time_end	Time Ended	Time	

Table Name: Transactions

Field	Description	Type	Length
trans_ID (PK)	Transaction ID	Int	11
trans_name	Transaction Name	Int	11
stud_ID (FK)	Student ID	Int	11
trans_date	Transaction Date	Date	

Table Name: Employee

Column Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(20)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)

Column Name	Null?	Type
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

4.2.2 Data Integrity and Constraints

It is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire lifecycle and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. The term is broad in scope and may have widely different meanings depending on the specific context – even under the same general umbrella of computing it is at times used as a proxy term for data quality while data validation is a pre-requisite for data integrity. Data integrity is the opposite of data corruption the overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities) and upon later retrieval, ensure the data is the same as it was when it was originally recorded. In short, data integrity aims to prevent unintentional changes to information. Data integrity is not to be confused with data, the discipline of protecting data from unauthorized parties.

Constraints are rules created at design-time that protect our data from becoming corrupt. It is essential for the long time survival of our heart child of a database solution. Without constraints solution will definitely decay with time and heavy usage. A wise man once said: "Data must protect itself!" And this is what constraints do. It is rules that keep the data in your database as valid as possible. There are many ways of doing this. Following are some of the ways to do so. Foreign key constraints are probably the most used constraint, and ensure that references to other tables are only allowed if there actually exists a target row to reference. This also makes it impossible to break such a relationship by deleting the referenced row creating a dead link.

4.3 Procedural design

Software Procedural Design (SPD) converts and translates structural elements into procedural explanations. SPD starts straight after data design and architectural design This has now been mostly abandoned mostly due to the rise in preference of object oriented programming and design pattern

The **procedural design** is often understood as a **software design** process that uses mainly control commands such as: sequence, condition, repetition, which are applied to the predefined data. Sequences serve to achieve the processing steps in order that is essential in the specification of any algorithm.

Example of Login Page:

A welcome screen will appear showing author's information like (name and password) as well as the file of the software from which the user will check

"Login".

A login screen will appears, then the user is required to type or Key in the

"user ID" and password".

From here, the program will open the main menu, where other sub menu are attached.

4.3.1 Logic Diagrams

Logic diagrams represent systematic flow of procedure that improves its comprehension and helps in implementation. Logic diagrams have many uses. In the solid state industry, they are used as the principal diagram for the design of solid state components such as computer chips.

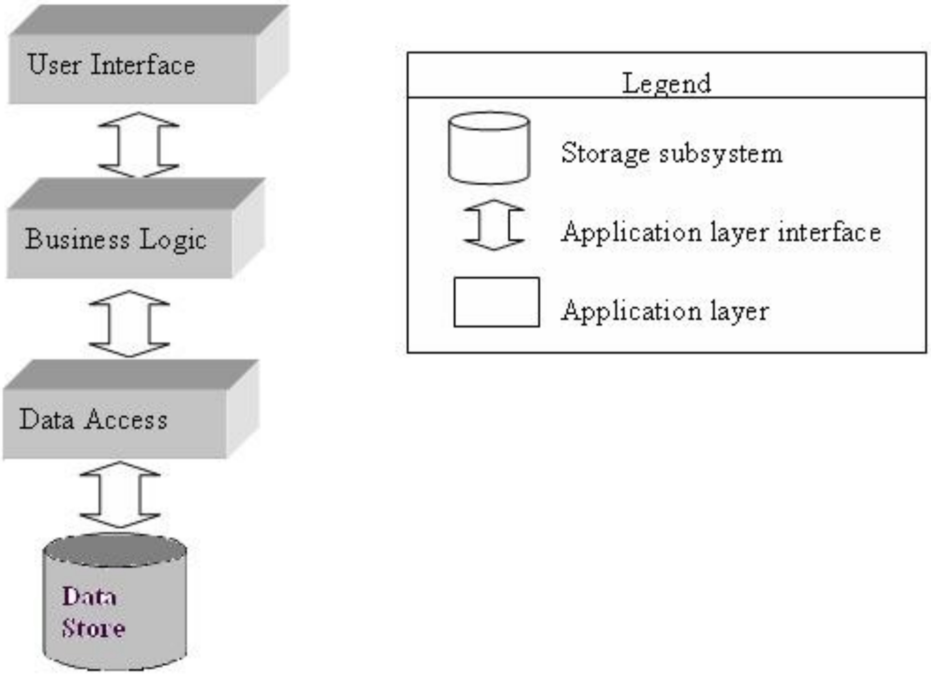


Figure- Logical Diagrams

A control flow diagram (CFD) is a diagram to describe the control flow of a business process, process or review. Network Diagrams aid in planning, organizing and controlling: Due to the sequential visualization of all project tasks and activities and all their dependencies, planning the project is an easier feat whilst being able to take into consideration the criticality of each task. Task interdependencies are clearly defined: With the help of visual representation of project tasks, their dependencies, criticality and duration are all clearly defined.

This allows for a more effective project workflow as team members receive a more in-depth understanding of the individual tasks and how to perform each one in order to reach the project's objective and goal. Figure 2 shows network diagram. It includes twenty activities. In activity diagram, workflow is defined which is important to have planned and understood. What activities are dependent on each other and what their sequence is all represented by network diagrams.

4.3.2 Data structure

In computer science, a **data structure** is a data organization, management and storage format that enable efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data

Data structures serve as the basis for abstract data type (ADT). "The ADT defines the logical form of the data type. The data structure implements the physical form of the data type.

Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, relational databases commonly use Btree indexes for data retrieval while Compiler implementations usually use hash tables to look up identifiers.

The implementation of a data structure usually requires writing a set of procedure that create and manipulate instances of that structure. The efficiency of a data structure cannot be analysed separately from those operations. This observation motivates the theoretical concept of an abstract data type, a data structure that is defined indirectly by the operations that may be performed on it, and the mathematical properties of those operations (including their space and time cost).

4.3.3 Algorithms Design

To make a computer do anything, we have to write a computer program. To write a computer program, we have to tell the computer, step by step, exactly what you want it to do. The computer then "executes" the program, following each step mechanically, to accomplish the end goal. When we are telling the computer what to do, we also get to choose how it's going to do it.

That's where computer algorithms come in. The algorithm is the basic technique used to get the job done. An algorithm is an unambiguous specification of how to solve a class of problems. It is a series of instructions, often referred to as a “process,” which is to be followed when solving a particular problem. It can perform calculation, data processing and automated reasoning tasks. In our project there are many algorithms for processing various types of data.

Login

1. START

2. Get Username and Password

3. If stored Username is equal to the entered Username & the stored Password is equal to the entered Password

4. Then login successful

5. Else login failed

6. End If.

7. END

```
START
  Get Username Get Password
        IF DATABASE EXIST
        THEN READ Username and Password
        FROM DATABASE
IF DATABASE. Username==Entered Username && DATABASE. Password ==
  Entered Password
                              Login successful
                              ELSE PRINT "Incorrect Username or
                              password" END IF
```

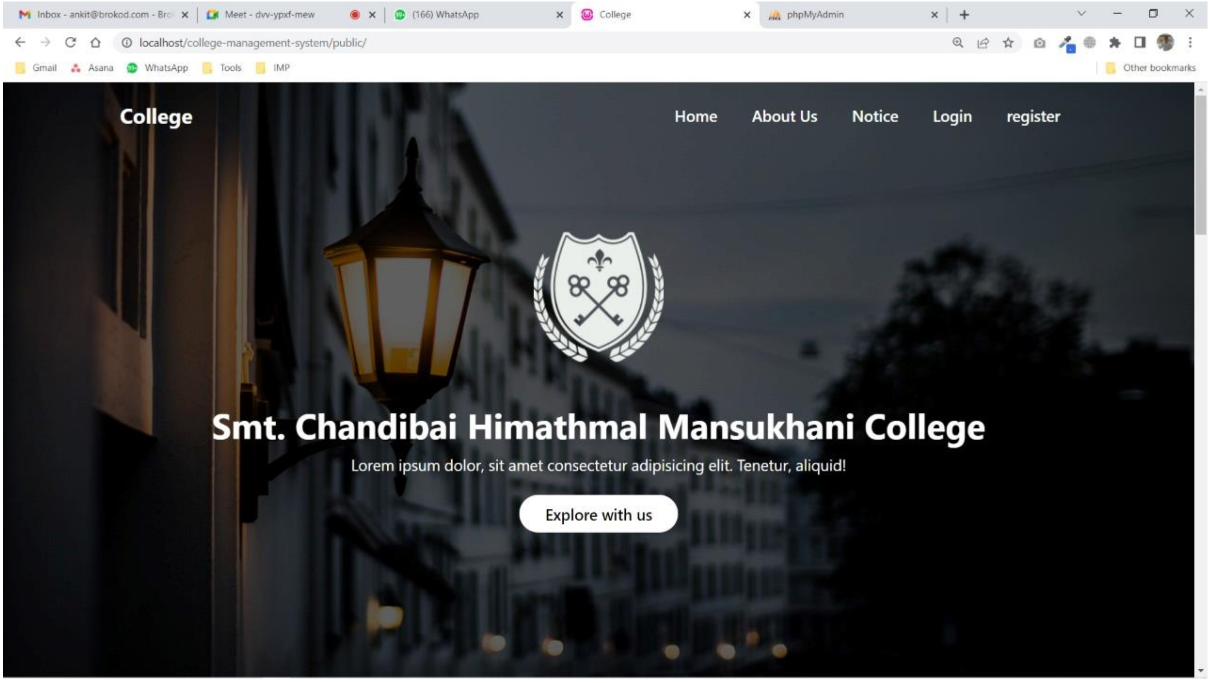
```
ELSE

        PRINT “DATABASE Error”

 END IF

 END
```

4.4 User interface

The **user interface** (**UI**), in the industrial field of human computer invention is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision making process. Examples of this broad concept of user interfaces include the interactive aspects of computer os, hand tools, heavy machinery operator controls, and controls. The design process considerations applicable when creating user interfaces are related to or involve such disciplines as ergonic and psychology

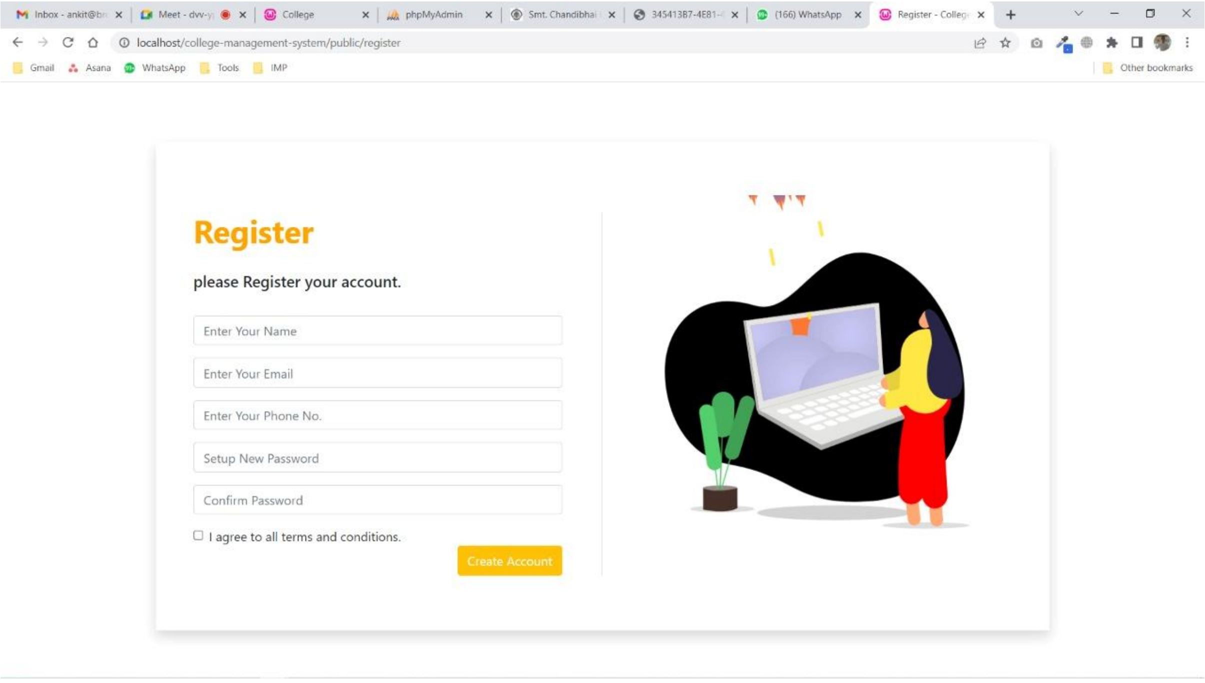
Generally, the goal of user interface is to produce a user interface which makes it easy, efficient, and enjoyable (user-friendly) to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human.

User interfaces are composed of one or more layers including a **human-machine interface** (**HMI**) interfaces machines with physical input hardware such a keyboards, mice, game pads and output hardware such as computer monitors, speakers, and printers. A device that implements a HMI is called a human interface device (HID). Other terms for human machine interfaces are **man–machine interface** (**MMI**) and when the machine in question is a computer **human–computer interface**. Additional UI layers may interact with one or more human sense, including: tactile UI (touch), visual UI (sight), auditory UI (sound), olfactory UI (smell), equilibrium UI (balance), and gustatory UI (taste).



Smt. Chandibai Himathmal Mansukhani College

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Tenetur, aliquid!



Register

please Register your account.

Figure-UI

4.5 Security Issues

Data is the most precious factor of today’s websites. Top business organizations spend billions of dollars every year to secure their computer networks and to keep their business data safe. We are dependent on computers today for controlling large money transfers between banks, insurance, markets, telecommunication, electrical power distribution, health and medical fields, nuclear power plants, space research and satellites. We cannot negotiate security in these critical areas.

Security is freedom from, or resilience against, potential harm (or other unwanted coercive change) from external forces. Beneficiaries (technically referents) of security may be persons and social groups, objects and institutions, ecosystems, and any other entity or phenomenon vulnerable to unwanted change by its environment. The most important service in today’s world is security. As technology is increasing, ways to hack data are also increasing. For any website or application development, security is one of the crucial aspects.

Security involves securing the data. In our system data is secured as only the person knowing username and password can enter the system. Passwords are stored in encrypted format. At time of registration, one security question is asked from users. If users forget their password at later stage they can access their account by answering that security question. This way system ensures that no other can use someone else’s account. The security involved in this system is hierarchical security. Admin is at top having all the rights.

4.6 Test Cases Design

Software testing is an essential and important technique for assessing the quality of a particular software product/service. In software testing, test cases and scenarios play an inevitable and a pivotal role. A good strategic design and technique help to improve the quality of the software testing process.the process improves the quality of the product/service and ensures effectiveness. Software testing is the process of analyzing a software item to know the differences between the existing and required conditions (bugs).Testing helps to evaluate the features of the software, to ensure it is free of bug. It is an activity that is carried out in co ordinance with the development cycle and before the deployment. A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Test cases	**Input specified**	**Expected result**	**Actual result**

User/admin	Username=””
	Password=””
		Username and password fields are empty.	Username and password fields are empty.

login	username=”abc”
	password=”abc123”
		Verify and then redirect to home page.	Verify and then redirect to home page.

	Email=””	Please enter email.	Please enter email.
Forgot password	Email=”abc”	Please enter valid email.	Please enter valid email.

	Email=”a2gzmail.com”	Email has been sent to your email-id.	Email has been sent to your email-id.
	Type=””	Please specify query type.	Please specify query type.
Queries type	Type=”MODIFY”	Please enter valid query type.	Please enter valid query type.
	Type=”CREATE”	Automatically creates CREATE query	Automatically creates CREATE query
User details	Name:”” Pin code:”” Addhar details:”” Address:””	Name,pincode, addhar details and address fields are empty	Name,pincode, addhar details and address fields are empty
	Name:”abc” Pincode:”421001” Addhardetails:”112233 445566” Address:” xyz ”	Verify and then redirect to home page.	Verify and then redirect to home page.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

During implementation, the project team creates the actual product. Product implementation can be an exciting phase for the customer, because their idea for the project becomes something tangible. Project developers begin building and coding the software. A software implementation method is a systematically structured approach to effectively integrate a software based service or component into the work flow of an organizational structure or an individual end-user. Implementation is not an event. It is a mission-oriented process involving multiple decisions, actions, and corrections designed to make full and effective use of effective innovations in education settings. For example, we might find that a new skill educators are using as part of social and emotional development could be further strengthened by additional coaching from an expert; so we would think about how to embed these strategies into ongoing coaching opportunities, and how we would gather data on if the coaching is leading to the improved use of this skills. Full Implementation – The skillful use of an innovation that is well-integrated into the repertoire of practitioners and routinely and effectively supported by successive program and local administrations, To implement a project means to carry out activities proposed in the application form with the aim to achieve project objectives and deliver results and outputs. Its success depends on many internal and external factors. Some of the most important ones are a very well organized project team and effective monitoring of project progress and related expenditures.

5.2 Code Details & Code Efficiency

Computer programming is the process of designing and building an executable computer program for accomplishing a specific computing task.

5.2.1 Code Details

Register.php:

```
<?php

include "../includes/common.php";

if(isset($_POST['register']))
{
    user_register();
}
?>

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Register - College Management System</title>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />

    <!-- Bootstrap CSS -->
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
      integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
      crossorigin="anonymous"
    />

    <link rel="stylesheet" href="../css/style.css" />
  </head>

  <body>
    <section class="gs-log">
      <div class="container shadow pr-5 pb-5 pl-5">

        <div class="row">
```

```
          <div class="col">
            <div class="text-center">
            <h4 class="text-center" style="font-weight: 700; padding-bottom: 20px; padding-top: 40px; color :orange"> <?php  display_message(); ?></h4>

            </div>
          </div>
        </div>

        <div class="row">
          <div class="col-lg-6 border-r">


            <h2>Register</h2>
            <h5 class="pb-4">
                   please Register your account.
            </h5>

            <div class="row">
              <div class="col">
                <form action="" method="post">
                <div class="form-group">
                  <input
                    type="text"
                    name="name"
                    class="form-control"
                    placeholder="Enter Your Name"
                    required
                    />
                </div>
              </div>
            </div>

            <div class="row">
              <div class="col">
                <div class="form-group">
                  <input
                    type="email"
                    name="email"
                    class="form-control"
                    placeholder="Enter Your Email"
                    required
                  />
                </div>
              </div>

            </div>

            <div class="row">
```

```
  <div class="col">
    <div class="form-group">
      <input
        type="text"
        name="phone"
        class="form-control"
        placeholder="Enter Your Phone No."
        required
      />
    </div>
  </div>
</div>



<div class="row">
  <div class="col">
    <div class="form-group">
      <input
        type="password"
        name="password"
        class="form-control"
        placeholder="Setup New Password"
        required
      />
    </div>
  </div>
</div>


<div class="row">
  <div class="col">
    <div class="form-group">
      <input
        type="password"
        name="c_password"
        class="form-control"
        placeholder="Confirm Password"
        required
      />
    </div>
  </div>
</div>



<div class="row">
```

```
              <div class="col">
                <div class="form-check">
                  <input
                    class="form-check-input"
                    type="checkbox"
                    value=""
                    id="flexCheckChecked"
                    required
                  />
                  <label class="form-check-label" for="flexCheckChecked">
                   I am agree to Lorem ipsum dolor sit, amet consectetur
adipisicing elit.
                  </label>
                </div>
              </div>
            </div>

            <div class="row">
              <div class="col">
                <div class="form-group">
                  <button
                    type="submit"
                    class="btn btn-warning text-white"
                    style="float: right"
                    name="register"
                  >
                    Create Account
                  </button>
                  </form>
                </div>
              </div>
            </div>
          </div>
          <div class="col-lg-6">
            <img
              src="https://brokod.github.io/assets/77628-website-launch.gif"
              class="img-fluid"
              alt=""
            />
          </div>
        </div>
      </div>
    </section>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script
      src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.
min.js"
      integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min
.js"
      integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

Login.php:

```
<?php

include "../includes/common.php";

if (isset($_POST['login'])) {
    user_login();
}
?>


<!DOCTYPE html>
<html lang="en">

<head>
    <title>Register - College Management System</title>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css
" integrity="sha384-
```

```
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous" />

    <link rel="stylesheet" href="../css/style.css" />
</head>

<body>
    <section class="gs-log">
        <div class="container shadow pr-5 pb-5 pl-5">

            <?php

            if (!empty(display_message())) {

            ?>

                <div class="row">
                    <div class="col">
                        <div class="text-center">
                            <h4 class="text-center" style="font-weight: 700;
padding-bottom: 20px; padding-top: 40px; color :orange"> <?php
display_message(); ?></h4>

                        </div>
                    </div>
                </div>
            <?php

            }
            ?>

            <div class="row">
                <div class="col-lg-6 border-r">


                    <h2>Login</h2>
                    <h5 class="pb-4">
                        Please Login To Continue Your Account.
                    </h5>


                    <div class="row">
                        <div class="col">
                            <div class="form-group">
                                <form action="" method="post">
                                <input type="email" name="email" class="form-
control" placeholder="Enter Your Email" required />
                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col">
                            <div class="form-group">
                                <input type="password" name="password"
class="form-control" placeholder="Enter Your Password" required />
                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col">
                            <div class="form-check">
                                <input class="form-check-input"
type="checkbox" value="" id="flexCheckChecked" required />
                                <label class="form-check-label"
for="flexCheckChecked">
                                    I am agree to Lorem ipsum dolor sit, amet
consectetur adipisicing elit.
                                </label>
                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col">
                            <div class="form-group">
                                <button type="submit" class="btn btn-warning
text-white" style="float: right" name="login">
                                    Login
                                </button>
                                </form>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-6">
                    <img src="https://brokod.github.io/assets/77628-website-
launch.gif" class="img-fluid" alt="" />
                </div>
            </div>
        </div>
    </section>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js
" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>

</html>
```

Index.php:

```
<?php include "includes/head.php";

$eemail = $_SESSION['email'];

$select_query = query("SELECT * FROM users WHERE email = '{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $name  = $row['name'];
        $email = $row['email'];
        $phone = $row['phone'];
        $phone2 = $row['phone2'];
    }
}

if (isset($_POST['basic_info_submit'])) {
    basic_info();
}

?>


<div class="container-fluid gs-cont">
    <div class="gs-container">
        <?php include "includes/sidebar.php"; ?>
        <div class="gs-body gs-col container-fluid bg-light" id="gs-body">
            <?php include "includes/nav.php"; ?>
            <div class="gs-content card m-3 p-3">



                <div class="row pt-2 ad-head">
                    <div class="col m-1">
                        <?php if (display_message()) {
                        ?>
                            <div class="bg-danger text-light  head"><?php
display_message(); ?></div>
                        <?php
                        }
                        ?>
                        <div class="bg-primary text-light  head">Basic
Information</div>
                        <div class="bg-light p-2 container-fluid">
                            <div class="row">

                                <div class="col-sm-6 col-xs-12">
                                    <form action="" method="post">
                                        <div class=" form-group">
                                            <label for="">
                                                Name:
                                            </label>
                                            <input type="text" name="name"
class="form-control" placeholder="Enter Your Name" value="<?php echo $name;
?>">
                                        </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Email:
                                        </label>
                                        <input type="email" name="email"
class="form-control" placeholder="Enter Your Email" value="<?php echo $email;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Phone:
                                        </label>
                                        <input type="text" name="phone" class="form-control" placeholder="Enter Your Phone No." value="<?php echo $phone; ?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Alternative Phone No.:
                                        </label>
                                        <input type="text" name="phone2" class="form-control" placeholder="Alternative Phone No." value="<?php echo $phone2; ?>">
                                    </div>
                                </div>

                                <div class="col-12 text-right">
                                    <div class=" form-group">
                                        <button type="submit" class="btn btn-primary" name="basic_info_submit">Submit & Next</button>
                                    </div>
                                    </form>
                                </div>

                            </div>
                        </div>
                    </div>
                </div>










            </div>
        </div>
    </div>
</div>
<?php include "includes/footer.php"; ?>
```

Personal-details.php:

```php
<?php include "includes/head.php";

$eemail = $_SESSION['email'];

$select_query = query("SELECT * FROM personal_details WHERE email =
'{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $email = $row['email'];
        $fname = $row['fname'];
        $mname = $row['mname'];
        $lname = $row['lname'];
        $pphone = $row['pphone'];
        $add_1 = $row['add_1'];
        $add_2 = $row['add_2'];
        $add_3 = $row['add_3'];
        $city = $row['city'];
        $pin_code = $row['pin_code'];
        $state = $row['state'];
        $country = $row['country'];
    }
}

$select_query = query("SELECT * FROM users WHERE email = '{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $name = $row['name'];
        $email = $row['email'];
        $phone2 = $row['phone2'];
    }
}

if (isset($_POST['personal_info_submit'])) {
    personal_info_submit();
}

?>


<div class="container-fluid gs-cont">
    <div class="gs-container">
        <?php include "includes/sidebar.php"; ?>
        <div class="gs-body gs-col container-fluid bg-light" id="gs-body">
```

```
            <?php include "includes/nav.php"; ?>
            <div class="gs-content card m-3 p-3">



                <div class="row pt-2 ad-head">
                    <div class="col m-1">



                        <?php if (display_message()) {
                        ?>
                            <div class="bg-danger text-light  head"><?php 
display_message(); ?></div>
                        <?php
                        }
                        ?>
                        <div class="bg-primary text-light  head">Personal 
Details </div>
                        <div class="bg-light p-2 container-fluid">
                            <div class="row">

                                <div class="col-sm-6 col-xs-12">
                                    <form action="" method="post">
                                        <div class=" form-group">
                                            <label for="">
                                                Name:
                                            </label>
                                            <input type="text" name="name" 
class="form-control" placeholder="Enter Your Name" value="<?php echo $name; 
?>" disabled>
                                        </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <form action="" method="post">
                                        <div class=" form-group">
                                            <label for="">
                                                Email:
                                            </label>
                                            <input type="email" class="form-
control" placeholder="Enter Your Email" value="<?php echo $email; ?>" 
disabled>
                                            <input type="hidden" name="email" 
class="form-control" value="<?php echo $email; ?>">
                                        </div>
                                </div>
                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Father Name:
                                        </label>
                                        <input type="text" name="fname"
class="form-control" placeholder="Enter Your Father Name" value="<?php echo
$fname; ?>">
                                    </div>
                                </div>
                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Mother Name:
                                        </label>
                                        <input type="text" name="mname"
class="form-control" placeholder="Enter Your Mother Name" value="<?php echo
$mname; ?>">
                                    </div>
                                </div>
                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Last Name:
                                        </label>
                                        <input type="text" name="lname"
class="form-control" placeholder="Enter Your Father Name" value="<?php echo
$lname; ?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Parents Phone No.:
                                        </label>
                                        <input type="text" name="pphone"
class="form-control" placeholder="Enter Your Parent Phone No." value="<?php
echo $pphone; ?>">
                                    </div>
                                </div>

                                <div class="col-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Address
                                        </label>
                                        <textarea name="add" id=""
class="form-control"><?php echo $add_1; ?></textarea>
                                    </div>
                                </div>

                                <div class="col-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Address 2
                                        </label>
                                        <textarea name="add2" id=""
class="form-control"><?php echo $add_2; ?></textarea>
                                    </div>
                                </div>

                                <div class="col-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Address 3
                                        </label>
                                        <textarea name="add3" id=""
class="form-control"><?php echo $add_3; ?></textarea>
                                    </div>
                                </div>


                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            City
                                        </label>
                                        <input type="text" name="city"
class="form-control" placeholder="Enter Your City" value="<?php echo $city;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            PIN
                                        </label>
                                        <input type="text" name="pin"
class="form-control" placeholder="Enter Your Pin" value="<?php echo $pin_code;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
```

```
                                    <div class=" form-group">
                                        <label for="">
                                            State
                                        </label>
                                        <input type="text" name="state"
class="form-control" placeholder="Enter Your State" value="<?php echo $state;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Country
                                        </label>
                                        <input type="text" name="country"
class="form-control" placeholder="Enter Your Country" value="<?php echo
$country; ?>">
                                    </div>
                                </div>





                                <div class="col-12 text-right">
                                    <div class=" form-group">
                                        <button type="submit" class="btn btn-
primary" name="personal_info_submit">Submit & Next</button>
                                    </div>
                                    </form>
                                </div>

                            </div>
                        </div>
                    </div>
                </div>









            </div>
        </div>
    </div>
```

```
</div>
<?php include "includes/footer.php"; ?>
```

Payments.php:

```
<?php include "includes/head.php";

$eemail = $_SESSION['email'];

$select_query = query("SELECT * FROM personal_details WHERE email =
'{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $email = $row['email'];
        $fname = $row['fname'];
    }
}

$select_query = query("SELECT * FROM users WHERE email = '{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $name = $row['name'];
        $email = $row['email'];
        $phone2 = $row['phone2'];
    }
}

if (isset($_POST['personal_info_submit'])) {
    personal_info_submit();
}

?>


<div class="container-fluid gs-cont">
    <div class="gs-container">
        <?php include "includes/sidebar.php"; ?>
        <div class="gs-body gs-col container-fluid bg-light" id="gs-body">
            <?php include "includes/nav.php"; ?>
            <div class="gs-content card m-3 p-3">
```

```
                <div class="row pt-2 ad-head">
                    <div class="col m-1">


                        <?php if (display_message()) {
                        ?>
                            <div class="bg-danger text-light  head"><?php
display_message(); ?></div>
                        <?php
                        }
                        ?>
                        <div class="bg-primary text-light  head">Payments
</div>
                        <div class="bg-light p-2 container-fluid">
                            <div class="row">

                            <div class="col-sm-12 col-xs-12 pt-2">
                            <form method="post"
action="../payment/paytm/pgRedirect.php">

                                        <div class=" form-group">
                                        <input id="ORDER_ID" type="hidden"
tabindex="1" maxlength="20" size="20"
                        name="ORDER_ID" autocomplete="off"
                        value="<?php echo  "ORDS" . rand(10000,99999999)?>">
                        <input type="hidden" title="TXN_AMOUNT" tabindex="10"
                        type="text" name="TXN_AMOUNT"
                        value="100">
                        <input type="hidden" id="CHANNEL_ID" tabindex="4"
maxlength="12"
                        size="12" name="CHANNEL_ID" autocomplete="off"
value="WEB">
                        <input type="hidden" id="INDUSTRY_TYPE_ID"
tabindex="4" maxlength="12" size="12" name="INDUSTRY_TYPE_ID"
autocomplete="off" value="Retail">

                        <input id="CUST_ID" tabindex="2"   type="hidden"
maxlength="12" size="12" name="CUST_ID" autocomplete="off" value="<?php
echo  rand(10000,99999999)?>">

                        <input type="checkbox" name="" class="form-checkbox"
id=""> <label for="">I hearby that i have accepted the all terms and
conditions</label>

                                    </div>
                                </div>
                                <div class="col-12 ">
                                    <div class=" form-group">
                                        <button type="submit" class="btn btn-
primary" name="personal_info_submit">Click Here To Pay</button>
                                    </div>
                                    </form>
                                </div>

                            </div>
                        </div>
                    </div>
                </div>









            </div>
        </div>
    </div>
</div>
<?php include "includes/footer.php"; ?>
```

Program.php

```
<?php include "includes/head.php";

$eemail = $_SESSION['email'];

$select_query = query("SELECT * FROM users WHERE email = '{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $name  = $row['name'];
        $email = $row['email'];
        $phone = $row['phone'];
        $phone2 = $row['phone2'];
    }
}
```

```
if (isset($_POST['basic_info_submit'])) {
    basic_info();
}

?>


<div class="container-fluid gs-cont">
    <div class="gs-container">
        <?php include "includes/sidebar.php"; ?>
        <div class="gs-body gs-col container-fluid bg-light" id="gs-body">
            <?php include "includes/nav.php"; ?>
            <div class="gs-content card m-3 p-3">



                <div class="row pt-2 ad-head">
                    <div class="col m-1">



                        <?php if (display_message()) {
                        ?>
                            <div class="bg-danger text-light  head"><?php
display_message(); ?></div>
                        <?php
                        }
                        ?>
                        <div class="bg-primary text-light  head">Select
Program And Campus  </div>
                        <div class="bg-light p-2 container-fluid">
                            <div class="row">

                                <div class="col-sm-6 col-xs-12">
                                    <form action="" method="post">
                                        <div class=" form-group">
                                            <label for="">
                                            PROGRAM APPLYING FOR :
                                            </label>
                                            <select name="program"  id=""
class="form-control">
                                                <option>Jr. College</option>
                                                <option>Sr. College</option>
                                            </select>
                                        </div>
                                </div>
                                <div class="col-12 text-right">
                                    <div class=" form-group">
                                        <!-- <button type="submit" class="btn
btn-primary" name="basic_info_submit">Submit & Next</button> -->
                                        <a href="personal-details" class="btn
btn-primary">Submit & Next</a>
                                    </div>
                                    </form>
                                </div>

                            </div>
                        </div>
                    </div>
                </div>











            </div>
        </div>
    </div>
</div>
<?php include "includes/footer.php"; ?>
```

Document-upload.php

```
<?php include "includes/head.php";

$eemail = $_SESSION['email'];

$select_query = query("SELECT * FROM personal_details WHERE email =
'{$eemail}' ");
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $email = $row['email'];
        $fname = $row['fname'];
    }
}

$select_query = query("SELECT * FROM users WHERE email = '{$eemail}' ");
```

```
confirm($select_query);

if (mysqli_num_rows($select_query) > 0) {
    while ($row = fetch_array($select_query)) {

        $name = $row['name'];
        $email = $row['email'];
        $phone2 = $row['phone2'];
    }
}

if (isset($_POST['personal_info_submit'])) {
    personal_info_submit();
}

?>


<div class="container-fluid gs-cont">
    <div class="gs-container">
        <?php include "includes/sidebar.php"; ?>
        <div class="gs-body gs-col container-fluid bg-light" id="gs-body">
            <?php include "includes/nav.php"; ?>
            <div class="gs-content card m-3 p-3">



                <div class="row pt-2 ad-head">
                    <div class="col m-1">



                        <?php if (display_message()) {
                        ?>
                            <div class="bg-danger text-light  head"><?php
display_message(); ?></div>
                        <?php
                        }
                        ?>
                        <div class="bg-primary text-light  head">Upload
Documents </div>
                        <div class="bg-light p-2 container-fluid">
                            <div class="row">


                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            SSC Marksheet
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            HSC Marksheet
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Adhar Card
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Pan Card
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Leaving Certificate
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
```

```
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Caste Certificate
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Domaciel Certificate
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Photo
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>

                                <div class="col-sm-6 col-xs-12">
                                    <div class=" form-group">
                                        <label for="">
                                            Signature
                                        </label>
                                        <input type="file" name="pphone"
class="form-control" placeholder="School Country." value="<?php echo $fname;
?>">
                                    </div>
                                </div>
                                <div class="col-12 text-right">
                                    <div class=" form-group">
                                        <button type="submit" class="btn btn-primary" name="personal_info_submit">Submit & Next</button>
                                    </div>
                                    </form>
                                </div>

                            </div>
                        </div>
                    </div>
                </div>









            </div>
        </div>
    </div>
</div>
<?php include "includes/footer.php"; ?>
```

Config_paytm.php

```
<?php
/*
- Use PAYTM_ENVIRONMENT as 'PROD' if you wanted to do transaction in production environment else 'TEST' for doing transaction in testing environment.
- Change the value of PAYTM_MERCHANT_KEY constant with details received from Paytm.
- Change the value of PAYTM_MERCHANT_MID constant with details received from Paytm.
- Change the value of PAYTM_MERCHANT_WEBSITE constant with details received from Paytm.
- Above details will be different for testing and production environment.
*/

define('PAYTM_ENVIRONMENT', 'TEST'); // PROD
define('PAYTM_MERCHANT_KEY', 'YpV4Sj!r&1TEh2pO'); //Change this constant's value with Merchant key received from Paytm.
define('PAYTM_MERCHANT_MID', 'fnyOGQ01849343850488'); //Change this constant's value with MID (Merchant ID) received from Paytm.
define('PAYTM_MERCHANT_WEBSITE', 'WEBSTAGING'); //Change this constant's value with Website name received from Paytm.

$PAYTM_STATUS_QUERY_NEW_URL='https://securegw-stage.paytm.in/merchant-status/getTxnStatus';
$PAYTM_TXN_URL='https://securegw-stage.paytm.in/theia/processTransaction';
if (PAYTM_ENVIRONMENT == 'PROD') {
    $PAYTM_STATUS_QUERY_NEW_URL='https://securegw.paytm.in/merchant-status/getTxnStatus';
    $PAYTM_TXN_URL='https://securegw.paytm.in/theia/processTransaction';
}

define('PAYTM_REFUND_URL', '');
define('PAYTM_STATUS_QUERY_URL', $PAYTM_STATUS_QUERY_NEW_URL);
define('PAYTM_STATUS_QUERY_NEW_URL', $PAYTM_STATUS_QUERY_NEW_URL);
define('PAYTM_TXN_URL', $PAYTM_TXN_URL);
?>
```

Pg_redirect.php

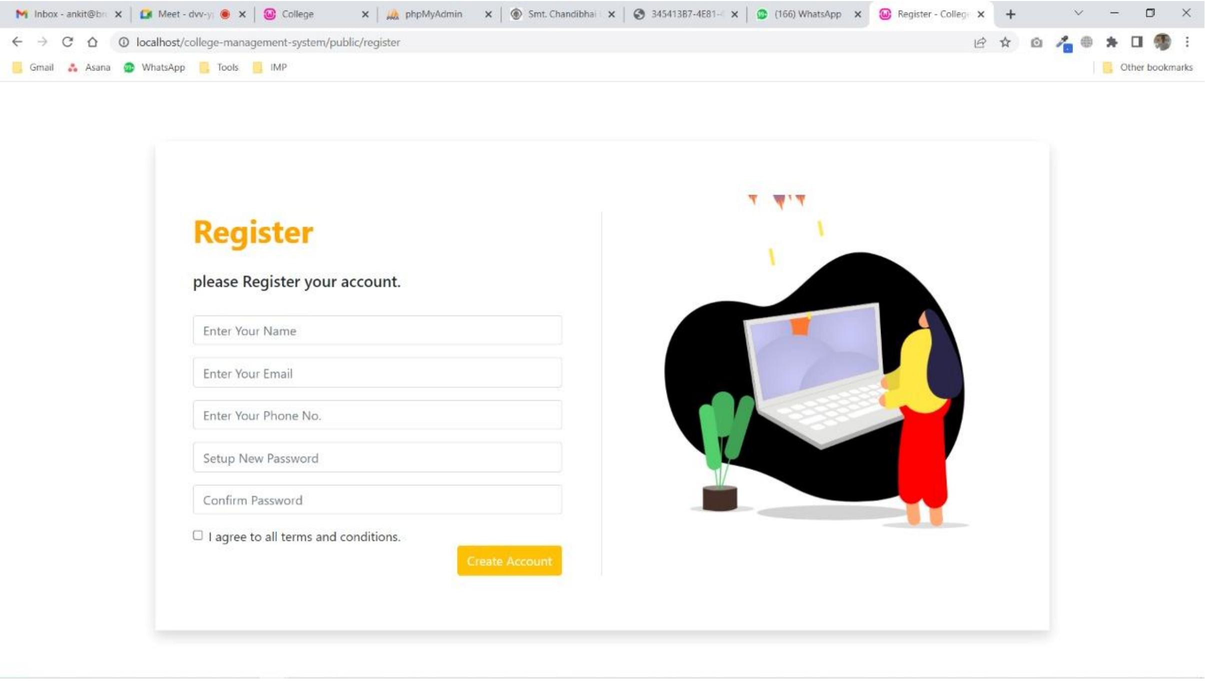
```
<?php
header("Pragma: no-cache");
header("Cache-Control: no-cache");
header("Expires: 0");
// following files need to be included
require_once("./lib/config_paytm.php");
require_once("./lib/encdec_paytm.php");

$checkSum = "";
$paramList = array();

$ORDER_ID = $_POST["ORDER_ID"];
$CUST_ID = $_POST["CUST_ID"];
$INDUSTRY_TYPE_ID = $_POST["INDUSTRY_TYPE_ID"];
$CHANNEL_ID = $_POST["CHANNEL_ID"];
$TXN_AMOUNT = $_POST["TXN_AMOUNT"];

// Create an array having all required parameters for creating checksum.
$paramList["MID"] = PAYTM_MERCHANT_MID;
$paramList["ORDER_ID"] = $ORDER_ID;
$paramList["CUST_ID"] = $CUST_ID;
$paramList["INDUSTRY_TYPE_ID"] = $INDUSTRY_TYPE_ID;
$paramList["CHANNEL_ID"] = $CHANNEL_ID;
$paramList["TXN_AMOUNT"] = $TXN_AMOUNT;
$paramList["WEBSITE"] = PAYTM_MERCHANT_WEBSITE;

/*
$paramList["CALLBACK_URL"] = "http://localhost/PaytmKit/pgResponse.php";
$paramList["MSISDN"] = $MSISDN; //Mobile number of customer
```

```
$paramList["EMAIL"] = $EMAIL; //Email ID of customer
$paramList["VERIFIED_BY"] = "EMAIL"; //
$paramList["IS_USER_VERIFIED"] = "YES"; //

*/

//Here checksum string will return by getChecksumFromArray() function.
$checkSum = getChecksumFromArray($paramList,PAYTM_MERCHANT_KEY);

?>
<html>
<head>
<title>Merchant Check Out Page</title>
</head>
<body>
    <center><h1>Please do not refresh this page...</h1></center>
        <form method="post" action="<?php echo PAYTM_TXN_URL ?>" name="f1">
        <table border="1">
            <tbody>
            <?php
            foreach($paramList as $name => $value) {
                echo '<input type="hidden" name="' . $name .'" value="' .
$value . '">';
            }
            ?>
            <input type="hidden" name="CHECKSUMHASH" value="<?php echo
$checkSum ?>">
            </tbody>
        </table>
        <script type="text/javascript">
            document.f1.submit();
        </script>
    </form>
</body>
</html>
```

Tnxstatus.php

```
<?php
    header("Pragma: no-cache");
    header("Cache-Control: no-cache");
    header("Expires: 0");

    // following files need to be included
    require_once("./lib/config_paytm.php");
    require_once("./lib/encdec_paytm.php");

    $ORDER_ID = "";
    $requestParamList = array();
    $responseParamList = array();
    if (isset($_POST["ORDER_ID"]) && $_POST["ORDER_ID"] != "") {

        // In Test Page, we are taking parameters from POST request. In actual
implementation these can be collected from session or DB.
        $ORDER_ID = $_POST["ORDER_ID"];

        // Create an array having all required parameters for status query.
        $requestParamList = array("MID" => PAYTM_MERCHANT_MID , "ORDERID" =>
$ORDER_ID);

        $StatusCheckSum =
getChecksumFromArray($requestParamList,PAYTM_MERCHANT_KEY);

        $requestParamList['CHECKSUMHASH'] = $StatusCheckSum;

        // Call the PG's getTxnStatusNew() function for verifying the
transaction status.
        $responseParamList = getTxnStatusNew($requestParamList);
    }

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Transaction status query</title>
<meta name="GENERATOR" content="Evrsoft First Page">
</head>
<body>
    <h2>Transaction status query</h2>
    <form method="post" action="">
        <table border="1">
            <tbody>
                <tr>
                    <td><label>ORDER_ID::*</label></td>
                    <td><input id="ORDER_ID" tabindex="1" maxlength="20"
size="20" name="ORDER_ID" autocomplete="off" value="<?php echo $ORDER_ID ?>">
                    </td>
                </tr>
                <tr>
                    <td></td>
                    <td><input value="Status Query"
type="submit"   onclick=""></td>
                </tr>
            </tbody>
        </table>
        <br/></br/>
        <?php
        if (isset($responseParamList) && count($responseParamList)>0 )
        {
        ?>
        <h2>Response of status query:</h2>
        <table style="border: 1px solid nopadding" border="0">
            <tbody>
                <?php
                    foreach($responseParamList as $paramName => $paramValue) {
                ?>
                <tr >
                    <td style="border: 1px solid"><label><?php echo
$paramName?></label></td>
                    <td style="border: 1px solid"><?php echo
$paramValue?></td>
                </tr>
                <?php
                    }
                ?>
            </tbody>
        </table>
        <?php
        }
        ?>
    </form>
</body>
</html>
```

Signup.php :



localhost/college-management-system/public/register

Register

please Register your account.

Enter Your Name

Enter Your Email

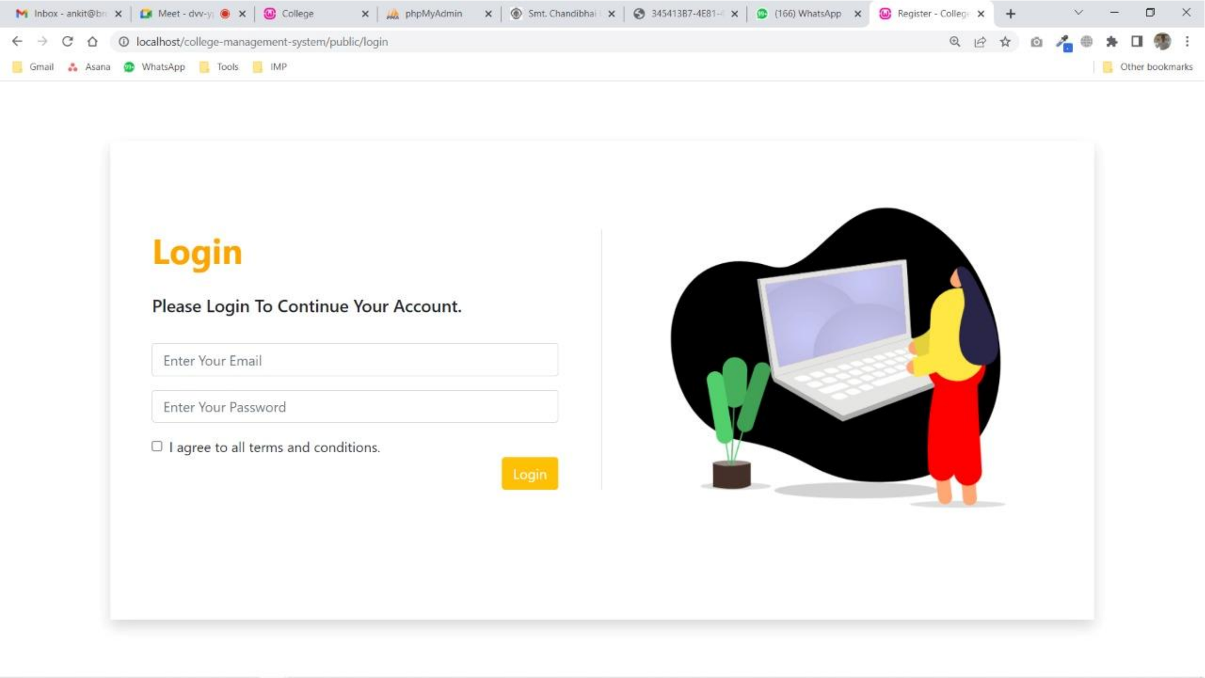
Enter Your Phone No.

Setup New Password

Confirm Password

□ I agree to all terms and conditions.

Login.php :



Login

Please Login To Continue Your Account.

Enter Your Email

Enter Your Password

□ I agree to all terms and conditions.

Login

5.2.2 Code Efficiency

Code efficiency is a broad term used to depict the reliability, speed and programming methodology used in developing codes for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software.

5.3 Test Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

5.3.1 Unit Testing

Unit Testing is the software development process in which the smallest unit of the system is tested. In C#, you can think of a unit as a method. You thus write a unit test by writing something that tests a method.

Unit Testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

5.3.2 Integration Testing

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules.

INTEGRATION TESTING is a type of software testing that combines and tests individual modules as a group. This level of testing is conducted to discover weaknesses in the interaction of integrated units.

Integration testing is aided by test drivers and test stubs. Integration testing is a sort of testing in which software elements are logically linked and tested as a unit.

A typical software project is made up of several software modules written by various programmers. Integration testing is concerned with ensuring that data is communicated between various units.

5.3.3 Beta Testing

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end-user (intended real user) validates the product for functionality, usability, reliability, and compatibility.

Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation.Since Beta Testing happens at the end user’s side, it cannot be the controlled activity.

5.4 Modifications and Improvement

Websites play a major role in the success of any business. Often it is the first point of contact For any new customer. If they like what they see online they are more likely to buy what you Have to sell. For an up to date website it should have the necessary website enhancement & modifications. Personalization of individual users has been in demand for years. Unfortunately, users want more. Today users want websites that are capable of predicting their behavior such as search suggestions based on previous interactions and browsing history. It is becoming increasingly important for users to be able to access websites even with limited connectivity. Unfortunately, the technology to offer this service is still growing

5.5 Test Cases

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Admin Test cases:

	Test scenario	Test case steps	Test case inputs Data	Expected results	Actual results
1	Admin login	1-Click on submit button without entering username & password		Enter your Username & Password	Same as expected result
		2-Entering wrong Username	Username=admin1 & Password=admin	Invalid username	Same as expected result

		3-Entering wrong Password	Username=admin & Password=admin1	Either username or password does not match	Same as expected result
		4-Entering correct password & email	Username=admin & Password=admin	Login successful	Same as expected result
2	Change password	1-click on submit button		Enter old password enter new password Retype password	Same as expected result
		2-Enter wrong old password	Old password=123 New=admin123 Retype New=admin123	Old password Does not Match. Please Enter it Again	Same as expected result
		3-Enter wrong retype password	Old password =123 New=admin123 Retype new=admin12	Password Does not Match	Same as expected result
		4-Enter right old password & correct retype password	Old password=123 New=admin123 Retype new=admin123	Password changed successfully	Same as expected result
3	Admin logout	1-Clickon Logout		Logout successfully	Same as expected result

4	Adding course details	1-Click on save button		Please enter the course name	Same as expected result
		2-Entering course name	Course name =Bsc IT	Please enter Character only	Same as expected result
5	Adding course details	1-click on save button		Enter course Name enter Select package	Same as expected result
		2-Entering numbers in course name	Course name=Mumbai	Please enter Character only	Same as expected result

Student Test cases:

	Test scenario	Test case steps	Test case inputs Data	Expected results	Actual results
1	Student login	1-Click on submit button without entering username & password		Enter your Username & Password	Same as expected result
		2-Entering wrong Username	Username=admin1 & Password=admin	Invalid username	Same as expected result

		3-Entering wrong Password	Username=admin & Password=admin1	Either username or password does not match	Same as expected result
		4-Entering correct password & email	Username=admin & Password=admin	Login successful	Same as expected result
2	Change password	1-click on submit button		Enter old password enter new password Retype password	Same as expected result
		2-Enter wrong old password	Old password=123 New=admin123 Retype New=admin123	Old password Does not Match. Please Enter it Again	Same as expected result
		3-Enter wrong retype password	Old password =123 New=admin123 Retype new=admin12	Password Does not Match	Same as expected result
		4-Enter right old password & correct retype password	Old password=123 New=admin123 Retype new=admin123	Password changed successfully	Same as expected result
3	Student logout	1-Clickon Logout		Logout successfully	Same as expected result

Chapter 6

RESULTS AND DISCUSSION

Document that records data obtained from an experiment of evaluation in an organized manner, describes the environmental or operating conditions, and shows the comparison of test results with test objectives.

6.1 Test Reports

This section covers the various tests, including validations that were conducted to ensure the proper working checking for the validity of the username of the Dynamic Timetable Generator.

Tests were also carried out to ensure that the proper error messages are displayed to the user whenever he/she is not authenticated. In case a user (admin or faculty) enters a wrong username and/or password, then a message box is displayed with an "Invalid Login" message is displayed.

Tests have also ensured that if a user logs into one faculty then same user cannot login to the application from any other machine without ending his/her previous login.

Tests were also conducted to ensure that the user gets the access to particular data which is only allowed to enter in the system.

Admin test cases Reports:

Test Case Number	Status	Result
1.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output
2.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output

3.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output
4.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output
5.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output

Customers Test Cases Reports:

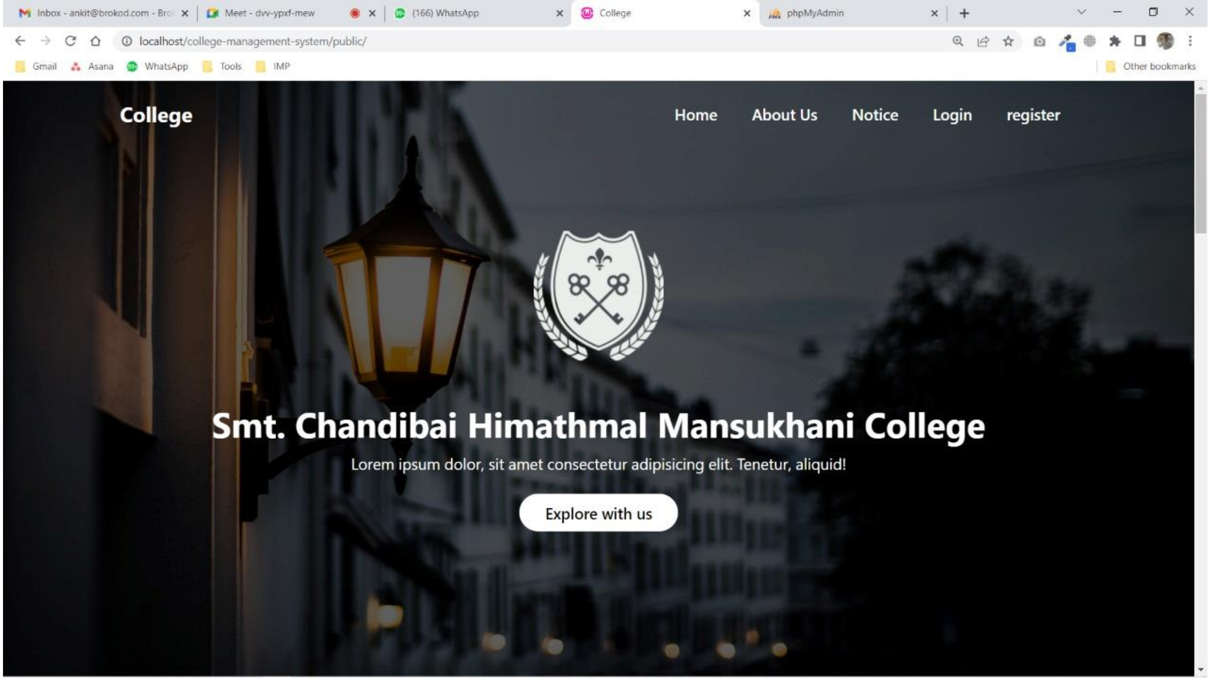
Test Case Number	Status	Result
1.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output
2.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output
3.	Completed	The result of all the inputs specification for this test case are passed and no exception is detected from actual output to expected output

6.2 User Documentation

The user documentation describes each feature of the program, and assists the user in realizing these features. A good user document can also go so far as to provide assistance. It is very important for user documents to not be confusing, and for them to be up to date.

Various guidance to users about our services.

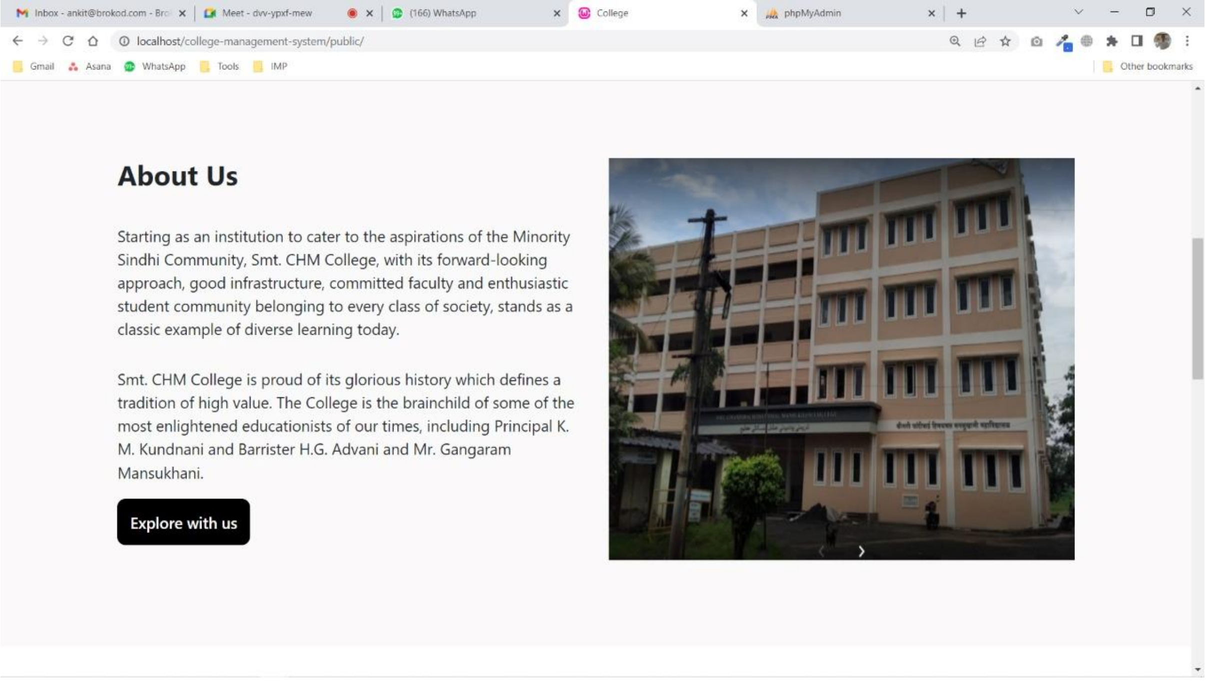
Homepage



Smt. Chandibai Himathmal Mansukhani College

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Tenetur, aliquid!

About us

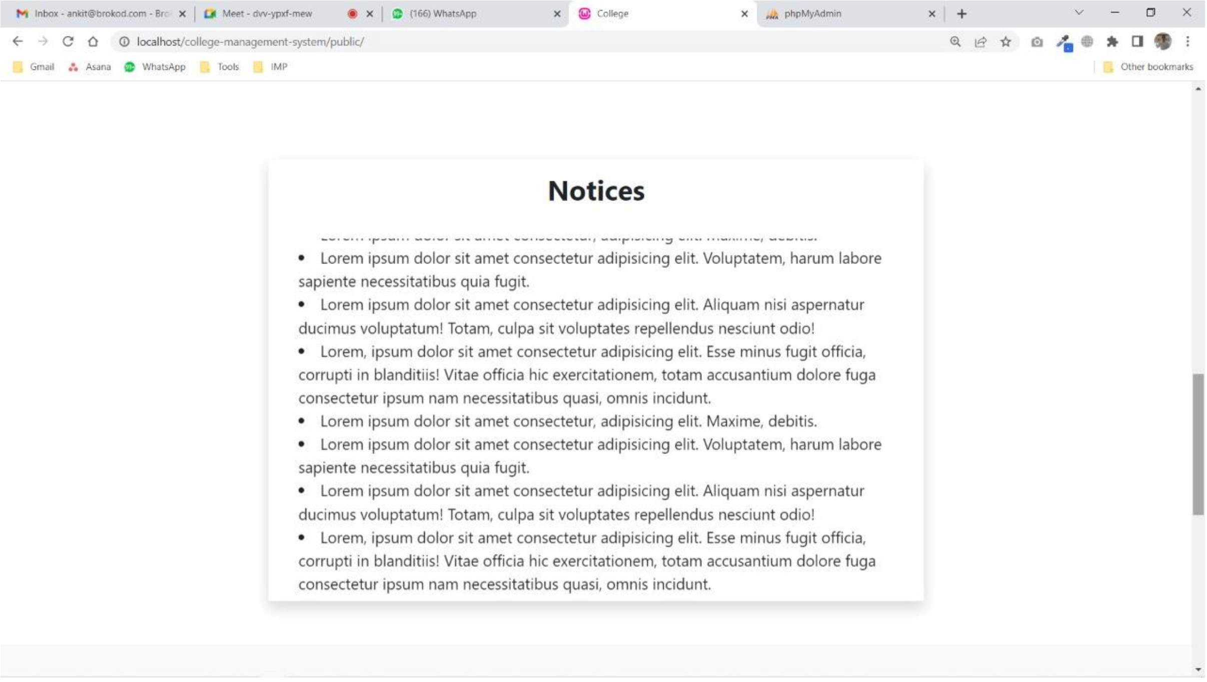


About Us

Starting as an institution to cater to the aspirations of the Minority Sindhi Community, Smt. CHM College, with its forward-looking approach, good infrastructure, committed faculty and enthusiastic student community belonging to every class of society, stands as a classic example of diverse learning today.

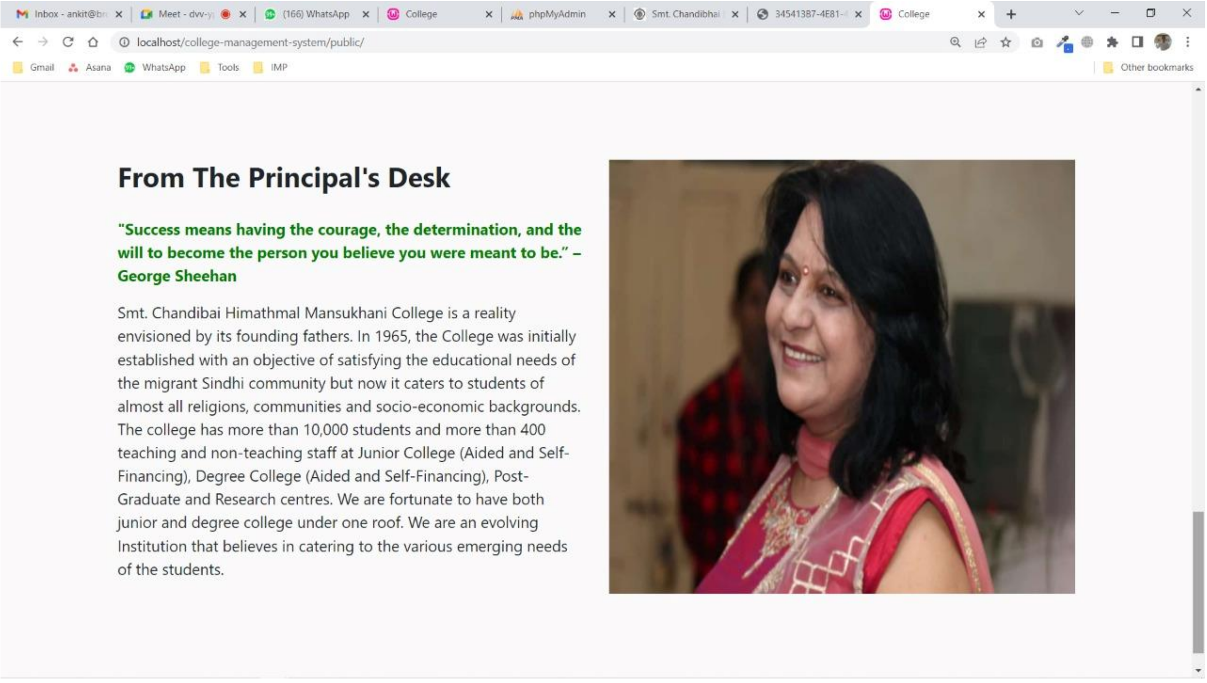
Smt. CHM College is proud of its glorious history which defines a tradition of high value. The College is the brainchild of some of the most enlightened educationists of our times, including Principal K. M. Kundnani and Barrister H.G. Advani and Mr. Gangaram Mansukhani.

Notice



Notices

- Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatem, harum labore sapiente necessitatibus quia fugit.
- Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam nisi aspernatur ducimus voluptatum! Totam, culpa sit voluptates repellendus nesciunt odio!
- Lorem, ipsum dolor sit amet consectetur adipisicing elit. Esse minus fugit officia, corrupti in blanditiis! Vitae officia hic exercitationem, totam accusantium dolore fuga consectetur ipsum nam necessitatibus quasi, omnis incidunt.
- Lorem ipsum dolor sit amet consectetur, adipisicing elit. Maxime, debitis.
- Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatem, harum labore sapiente necessitatibus quia fugit.
- Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam nisi aspernatur ducimus voluptatum! Totam, culpa sit voluptates repellendus nesciunt odio!
- Lorem, ipsum dolor sit amet consectetur adipisicing elit. Esse minus fugit officia, corrupti in blanditiis! Vitae officia hic exercitationem, totam accusantium dolore fuga consectetur ipsum nam necessitatibus quasi, omnis incidunt.

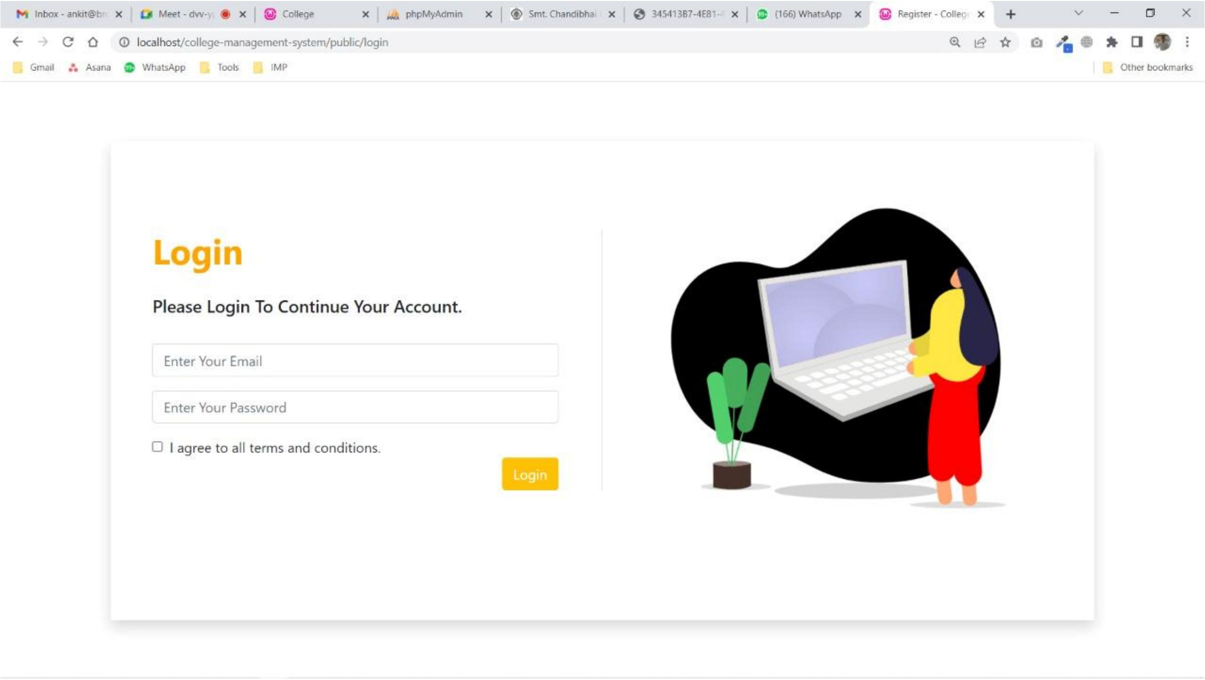


From The Principal's Desk

"Success means having the courage, the determination, and the will to become the person you believe you were meant to be." – George Sheehan

Smt. Chandibai Himathmal Mansukhani College is a reality envisioned by its founding fathers. In 1965, the College was initially established with an objective of satisfying the educational needs of the migrant Sindhi community but now it caters to students of almost all religions, communities and socio-economic backgrounds. The college has more than 10,000 students and more than 400 teaching and non-teaching staff at Junior College (Aided and Self-Financing), Degree College (Aided and Self-Financing), Post-Graduate and Research centres. We are fortunate to have both junior and degree college under one roof. We are an evolving Institution that believes in catering to the various emerging needs of the students.

Login



Login

Please Login To Continue Your Account.

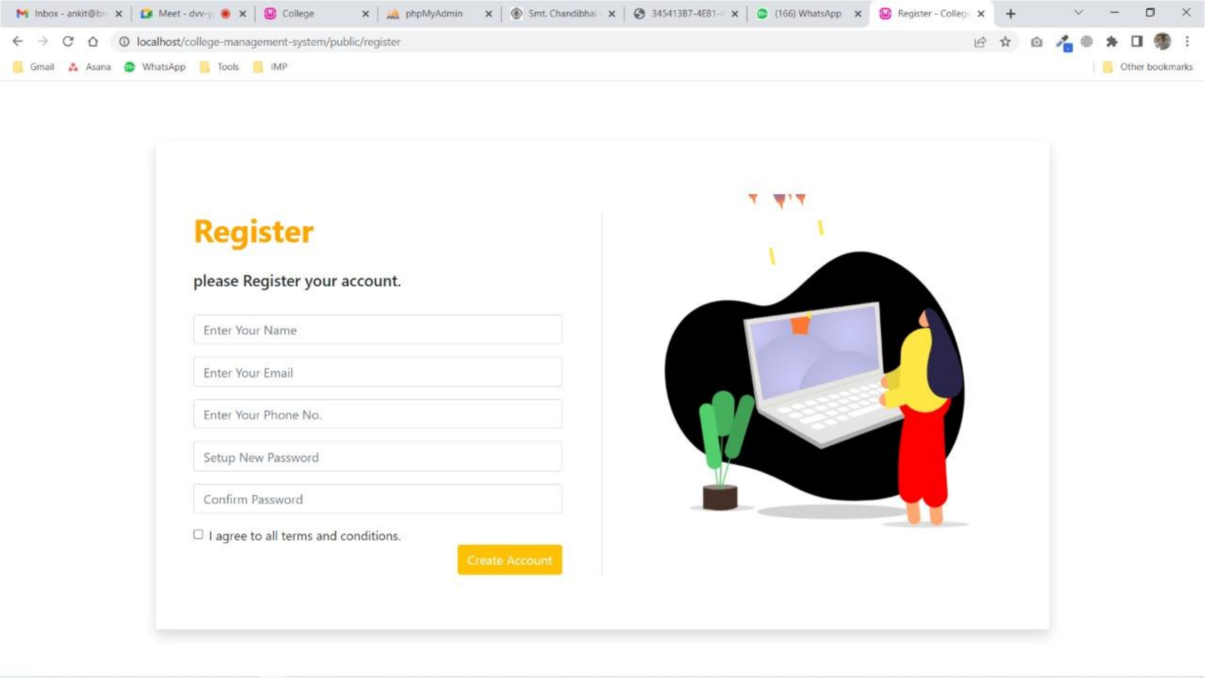
Enter Your Email

Enter Your Password

□ I agree to all terms and conditions.

Login

Register



Register

please Register your account.

Enter Your Name

Enter Your Email

Enter Your Phone No.

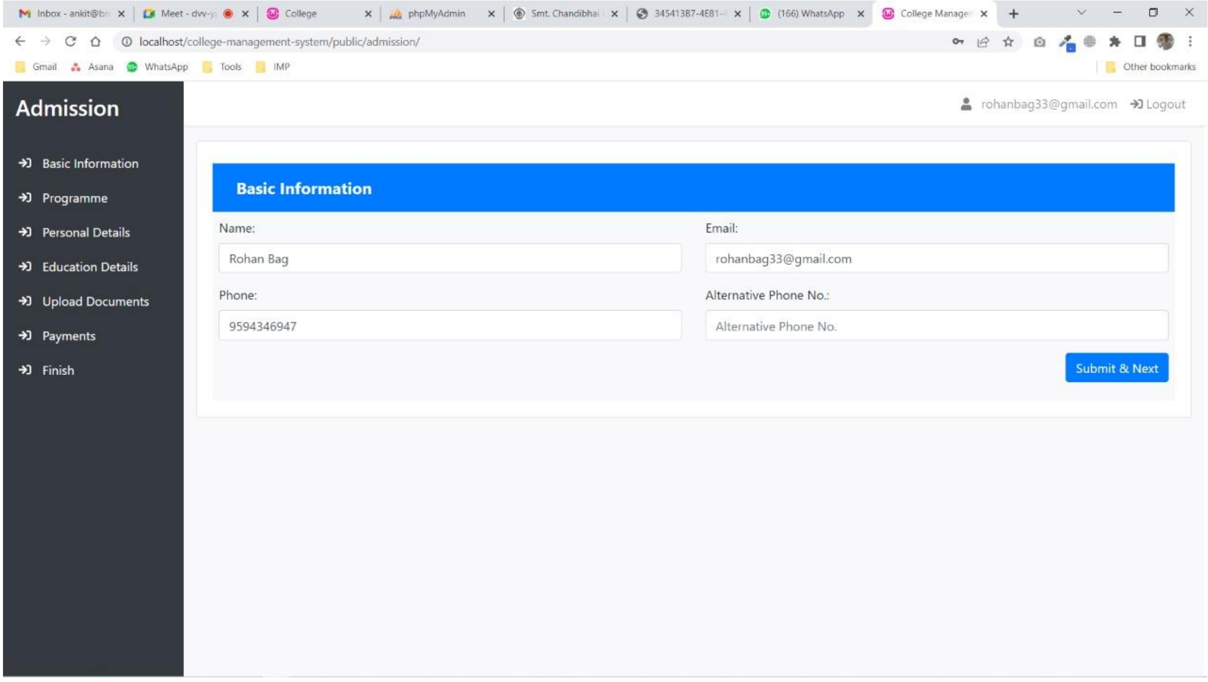
Setup New Password

Confirm Password

□ I agree to all terms and conditions.

Create Account

Admission form



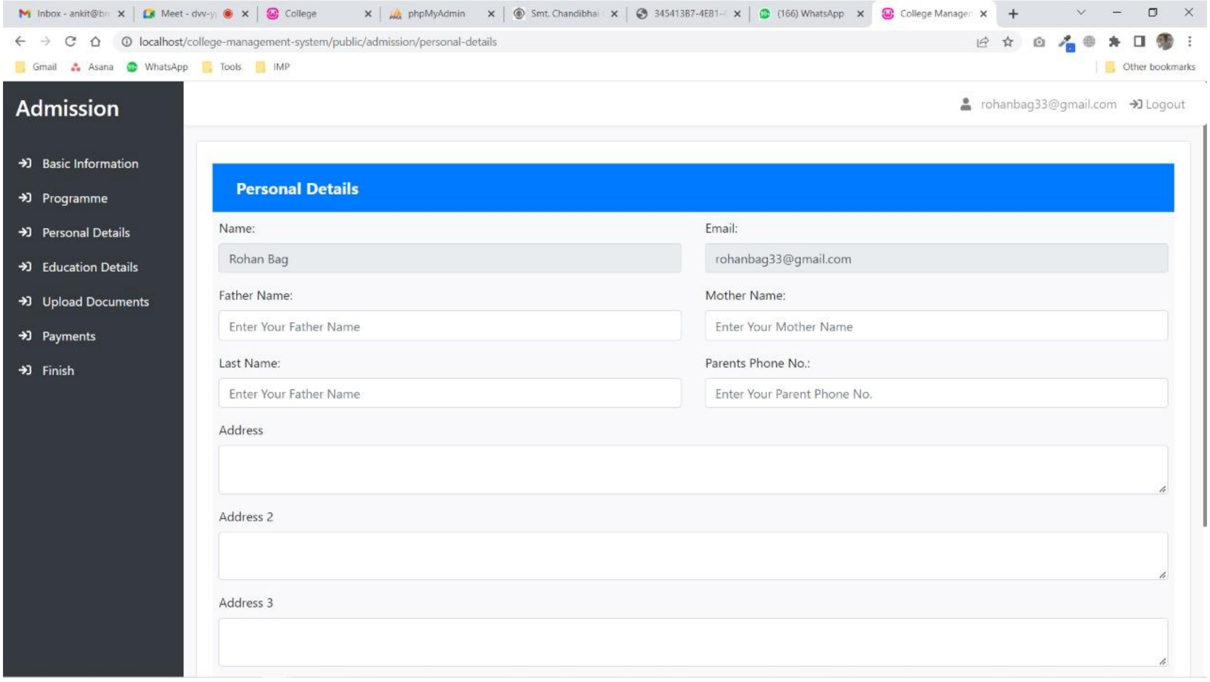
Basic Information

Name: Rohan Bag

Email: rohanbag33@gmail.com

Phone: 9594346947

Alternative Phone No.: Alternative Phone No.



Personal Details

Name: Rohan Bag

Email: rohanbag33@gmail.com

Father Name: Enter Your Father Name

Mother Name: Enter Your Mother Name

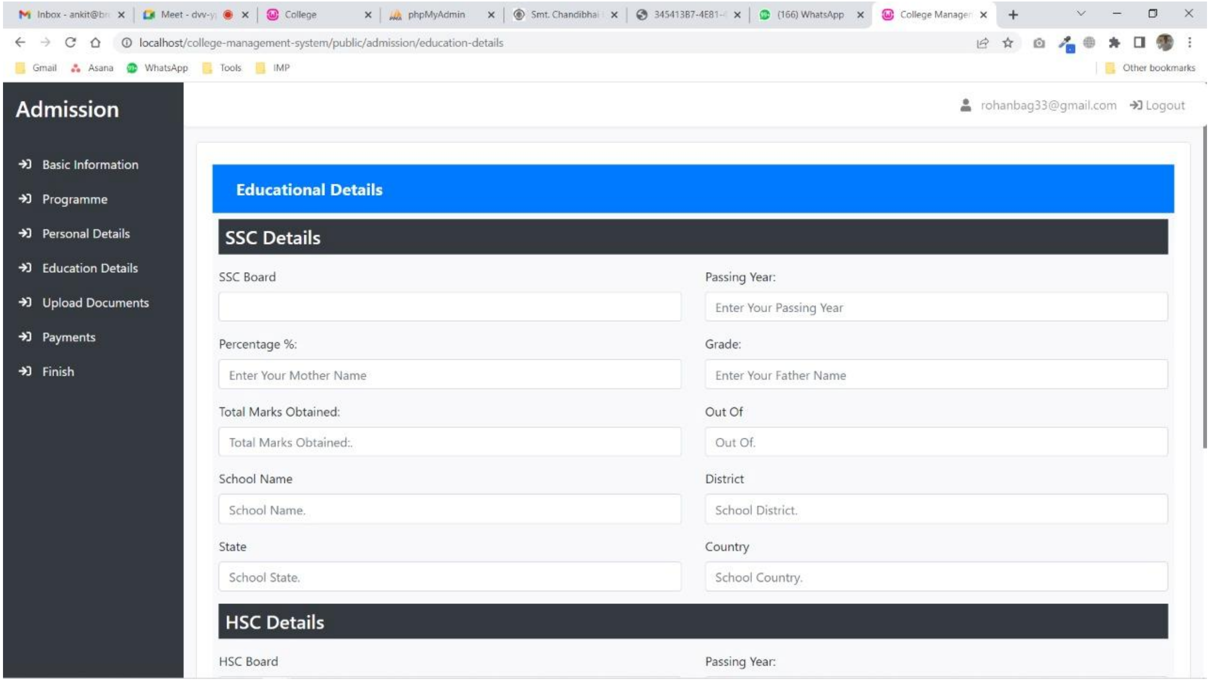
Last Name: Enter Your Father Name

Parents Phone No.: Enter Your Parent Phone No.

Address

Address 2

Address 3



Admission

- Basic Information
- Programme
- Personal Details
- Education Details
- Upload Documents
- Payments
- Finish

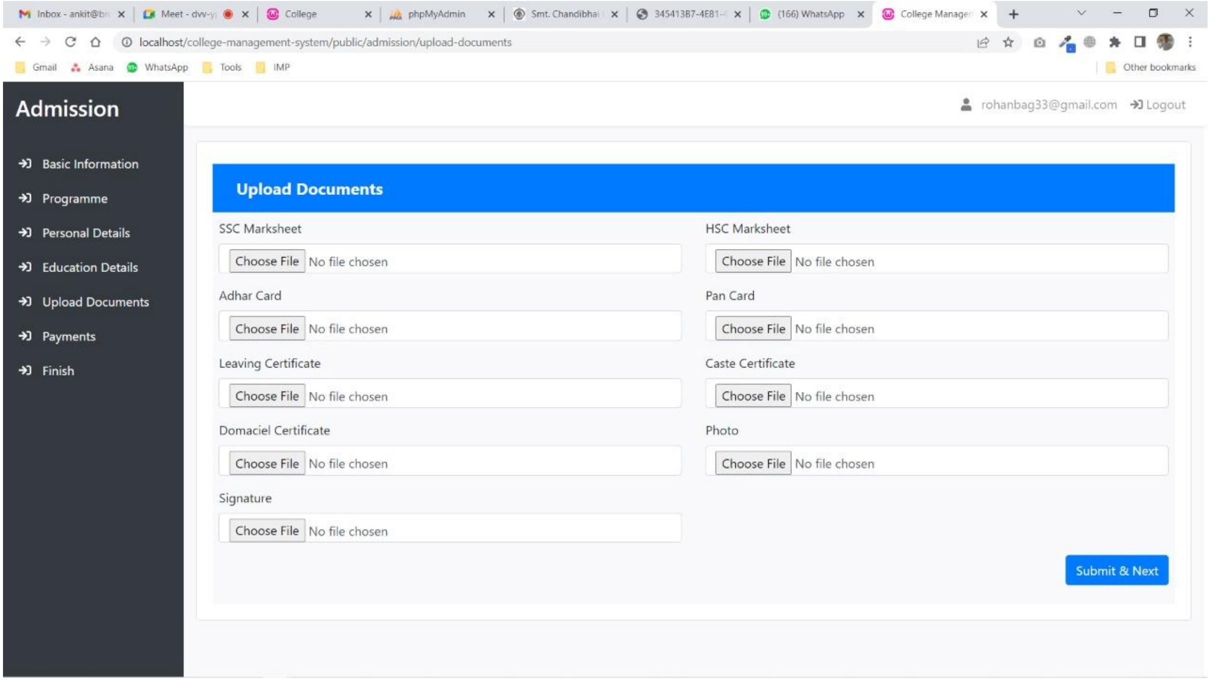
Educational Details

SSC Details

SSC Board	Passing Year:
	Enter Your Passing Year
Percentage %:	Grade:
Enter Your Mother Name	Enter Your Father Name
Total Marks Obtained:	Out Of
Total Marks Obtained:.	Out Of.
School Name	District
School Name.	School District.
State	Country
School State.	School Country.

HSC Details

HSC Board	Passing Year:

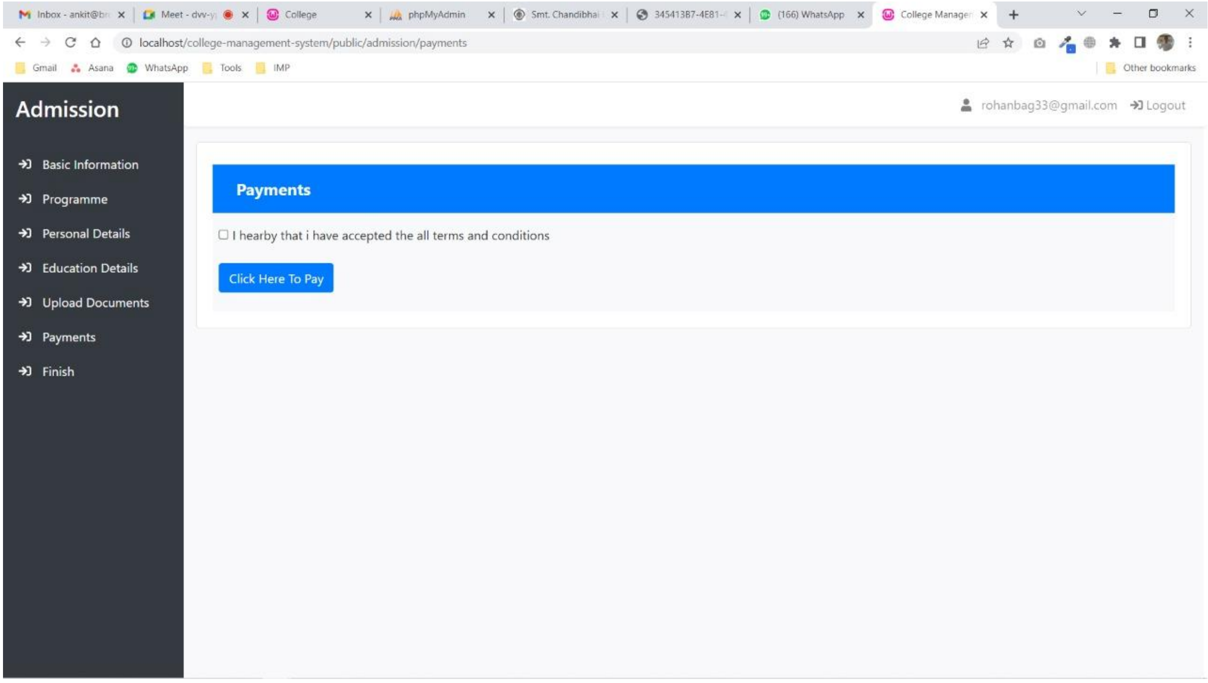


Admission

- Basic Information
- Programme
- Personal Details
- Education Details
- Upload Documents
- Payments
- Finish

Upload Documents

SSC Marksheet	HSC Marksheet
Choose File No file chosen	Choose File No file chosen
Adhar Card	Pan Card
Choose File No file chosen	Choose File No file chosen
Leaving Certificate	Caste Certificate
Choose File No file chosen	Choose File No file chosen
Domaciel Certificate	Photo
Choose File No file chosen	Choose File No file chosen
Signature	
Choose File No file chosen	

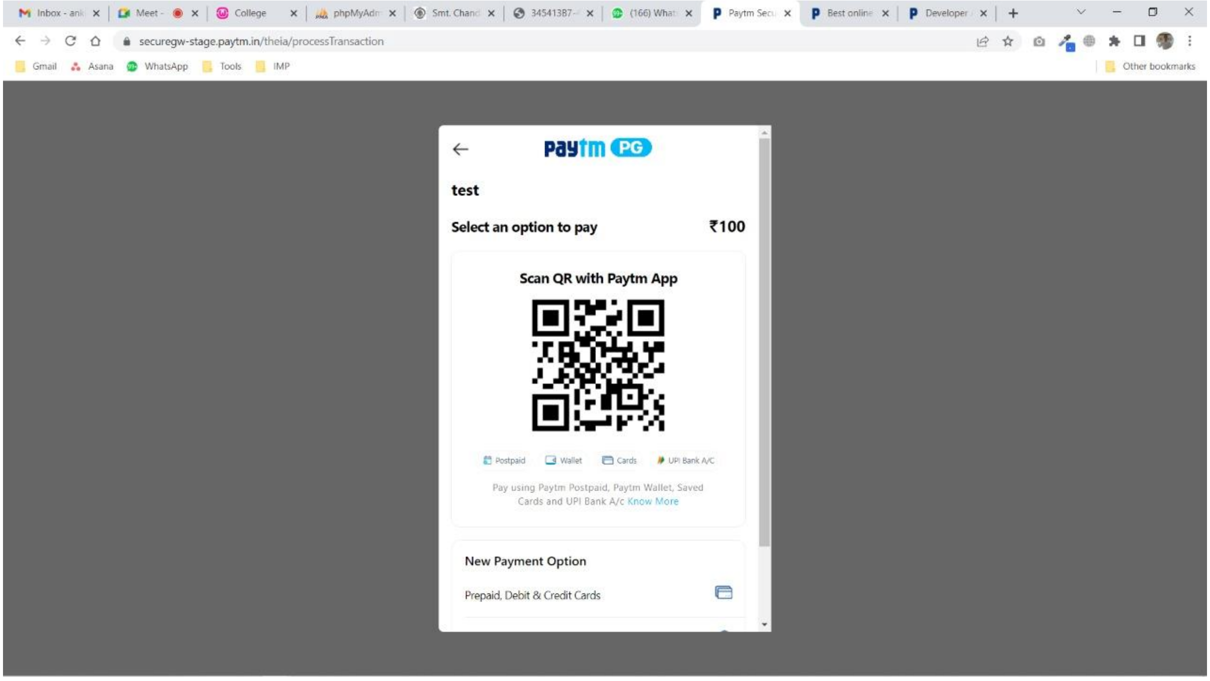


Payments

□ I hearby that i have accepted the all terms and conditions

Click Here To Pay

Payment



test

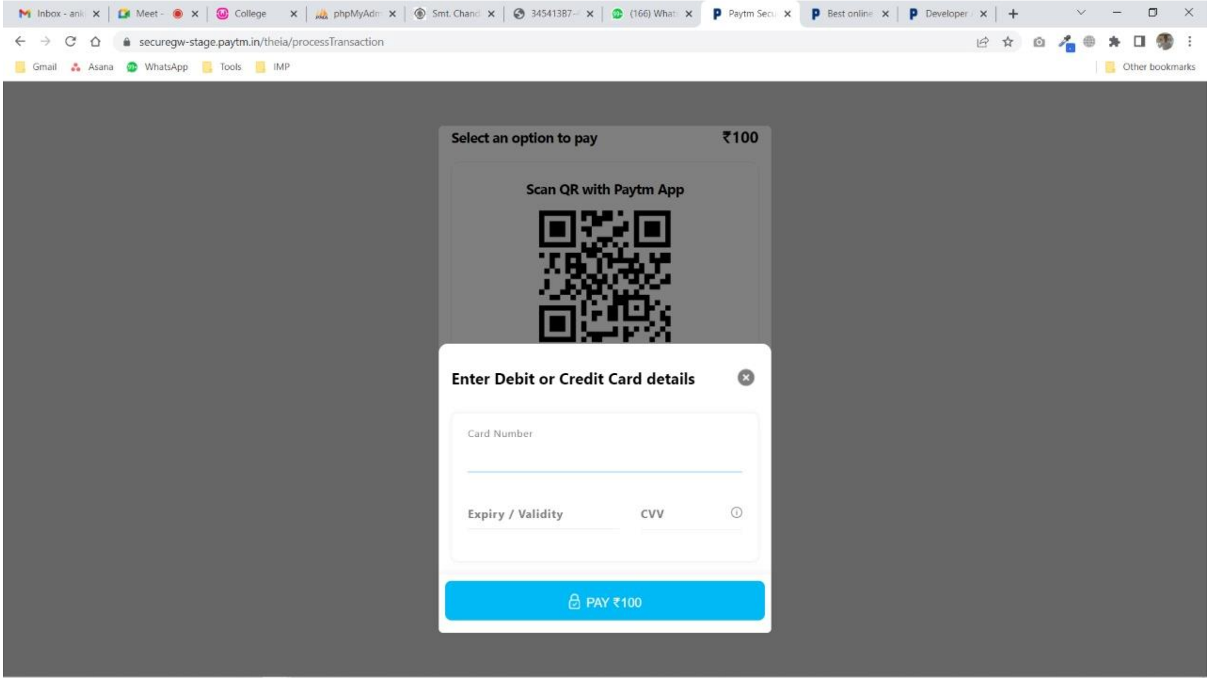
Select an option to pay ₹100

Scan QR with Paytm App

Pay using Paytm Postpaid, Paytm Wallet, Saved Cards and UPI Bank A/c Know More

New Payment Option

Prepaid, Debit & Credit Cards



Select an option to pay ₹100

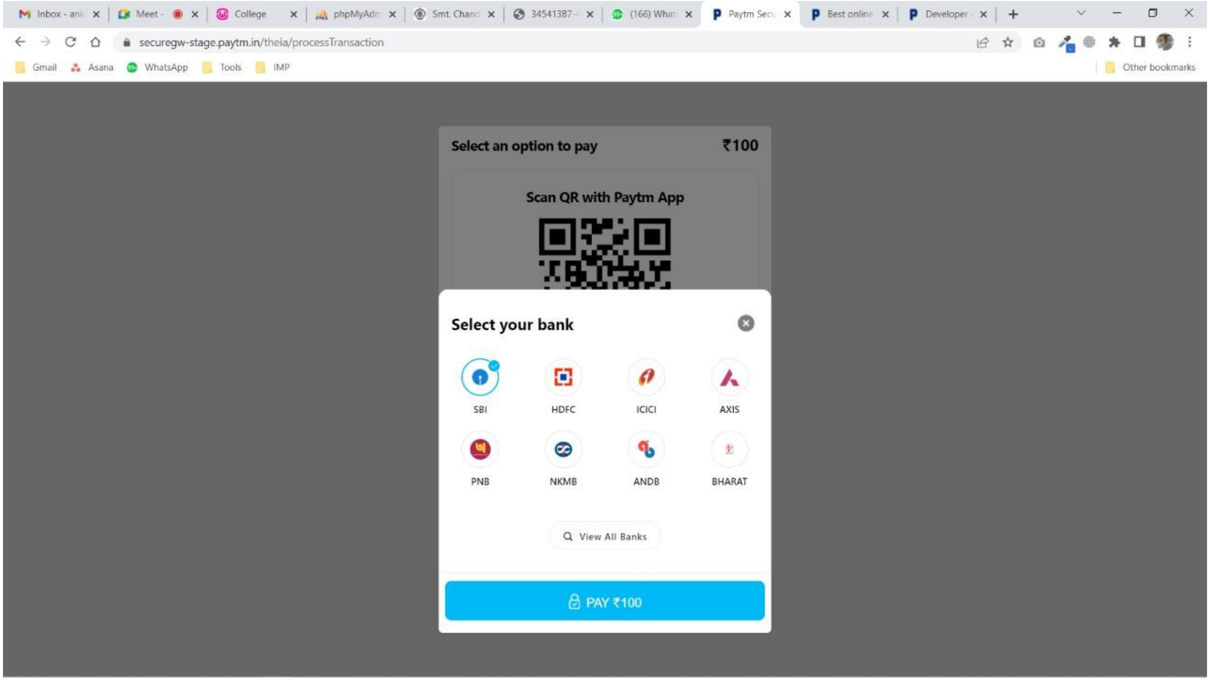
Scan QR with Paytm App

Enter Debit or Credit Card details

Card Number

Expiry / Validity

CVV



Select an option to pay ₹100

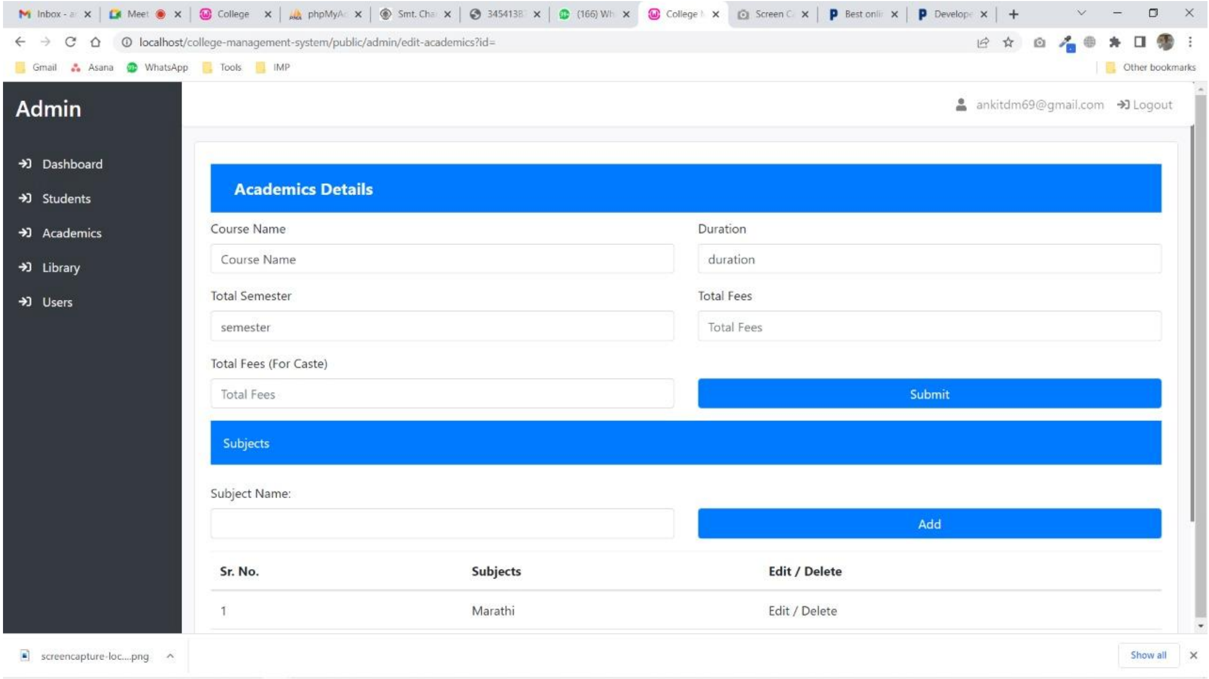
Scan QR with Paytm App

Select your bank

SBI HDFC ICICI AXIS

PNB NKMB ANDB BHARAT

Admin



Academics Details

Course Name: Course Name

Duration: duration

Total Semester: semester

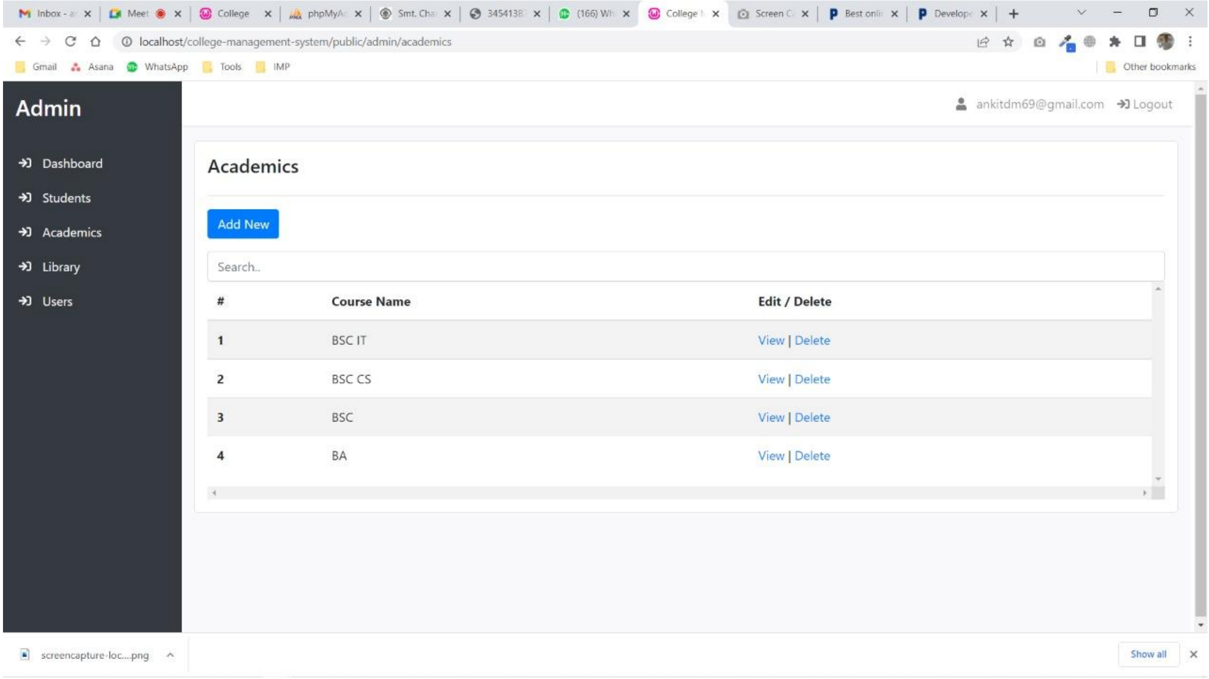
Total Fees: Total Fees

Total Fees (For Caste): Total Fees

Subjects

Subject Name:

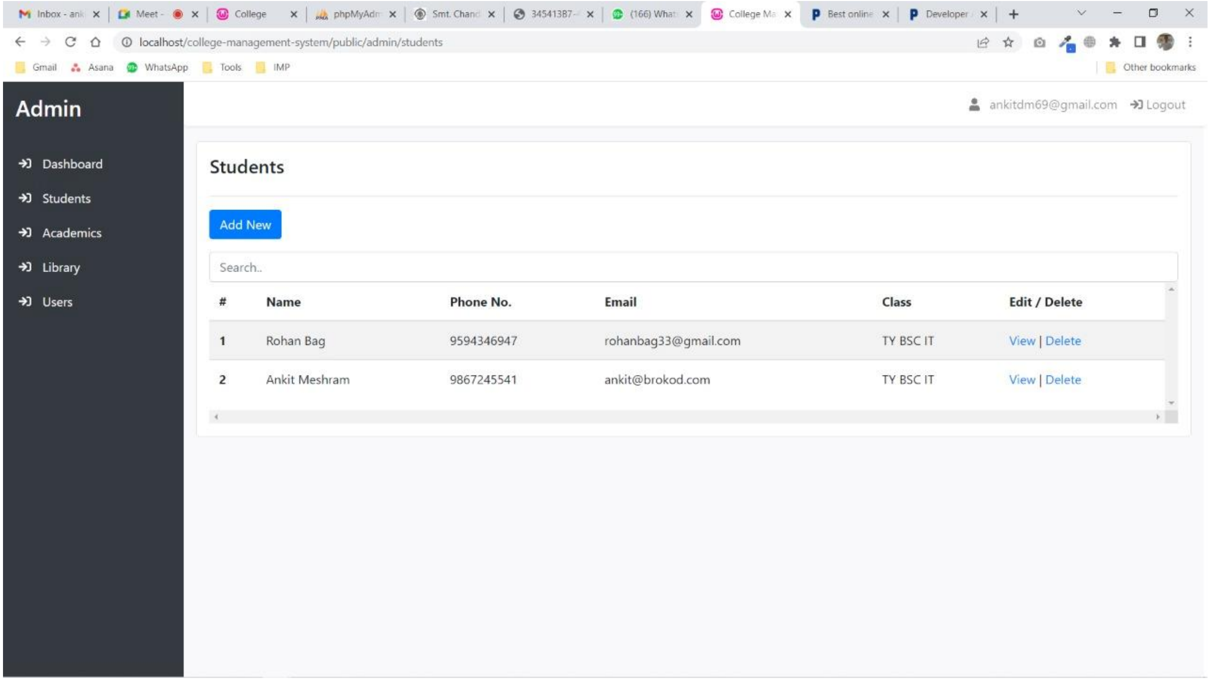
Sr. No.	Subjects	Edit / Delete
1	Marathi	Edit / Delete



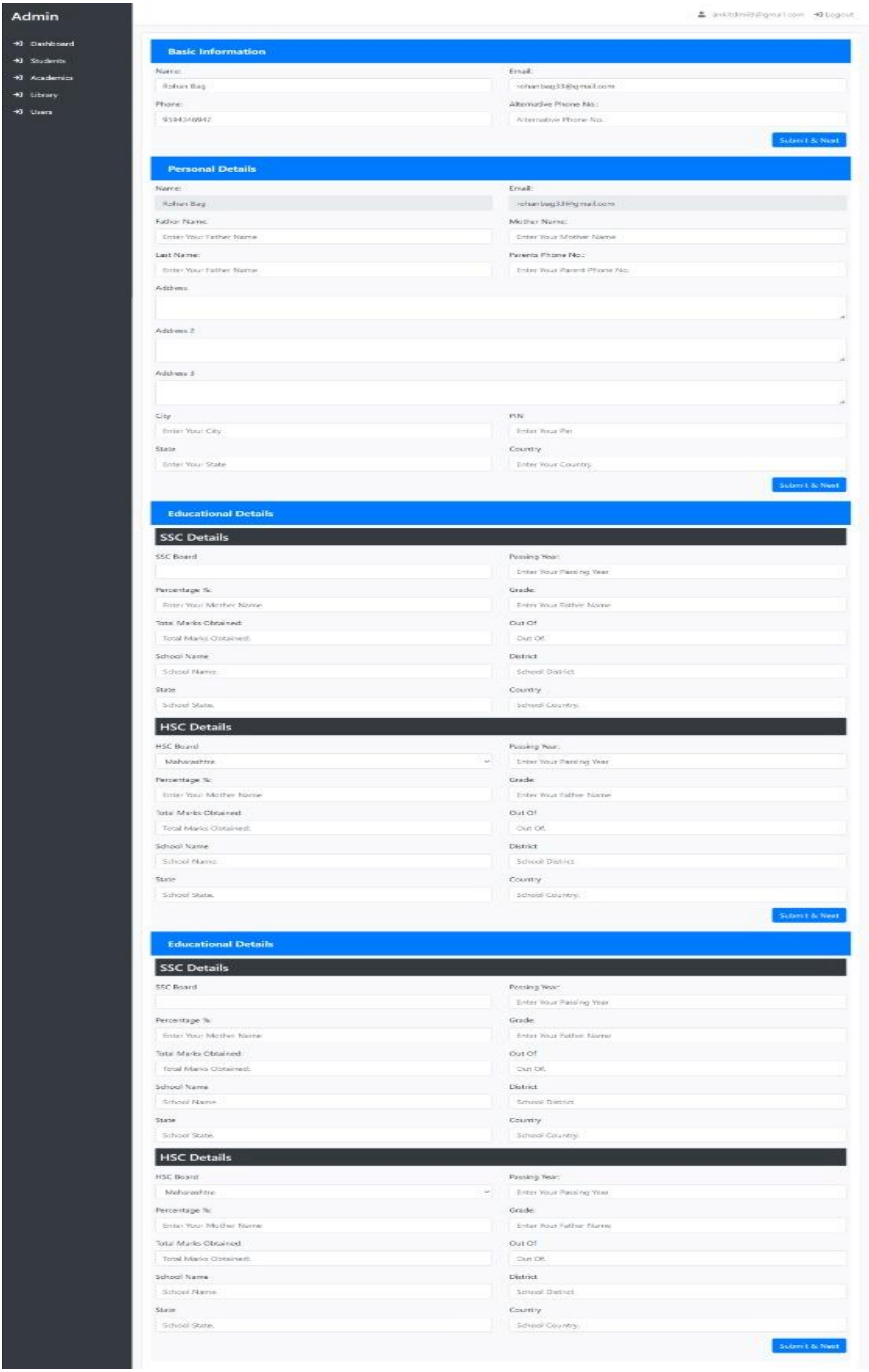
Academics

#	Course Name	Edit / Delete
1	BSC IT	View | Delete
2	BSC CS	View | Delete
3	BSC	View | Delete
4	BA	View | Delete

Student manager



#	Name	Phone No.	Email	Class	Edit / Delete
1	Rohan Bag	9594346947	rohanbag33@gmail.com	TY BSC IT	View | Delete
2	Ankit Meshram	9867245541	ankit@brokod.com	TY BSC IT	View | Delete



Basic Information

Name:
Rohan Bag

Email:
rohanbag33@gmail.com

Phone:
9594346947

Alternative Phone No.:
Alternative Phone No.

Personal Details

Name:
Rohan Bag

Email:
rohanbag33@gmail.com

Father Name:
Enter Your Father Name

Mother Name:
Enter Your Mother Name

Last Name:
Enter Your Father Name

Parents Phone No.:
Enter Your Parent Phone No.

Address

Address 2

Address 3

City
Enter Your City

PIN
Enter Your Pin

State
Enter Your State

Country
Enter Your Country

Educational Details

SSC Details

SSC Board

Passing Year:
Enter Your Passing Year

Percentage %:
Enter Your Mother Name

Grade:
Enter Your Father Name

Total Marks Obtained:
Total Marks Obtained

Out Of
Out Of

School Name
School Name

District
School District

State
School State

Country
School Country

HSC Details

HSC Board
Maharashtra

Passing Year:
Enter Your Passing Year

Percentage %:
Enter Your Mother Name

Grade:
Enter Your Father Name

Total Marks Obtained:
Total Marks Obtained

Out Of
Out Of

School Name
School Name

District
School District

State
School State

Country
School Country

Educational Details

SSC Details

SSC Board

Passing Year:
Enter Your Passing Year

Percentage %:
Enter Your Mother Name

Grade:
Enter Your Father Name

Total Marks Obtained:
Total Marks Obtained

Out Of
Out Of

School Name
School Name

District
School District

State
School State

Country
School Country

HSC Details

HSC Board
Maharashtra

Passing Year:
Enter Your Passing Year

Percentage %:
Enter Your Mother Name

Grade:
Enter Your Father Name

Total Marks Obtained:
Total Marks Obtained

Out Of
Out Of

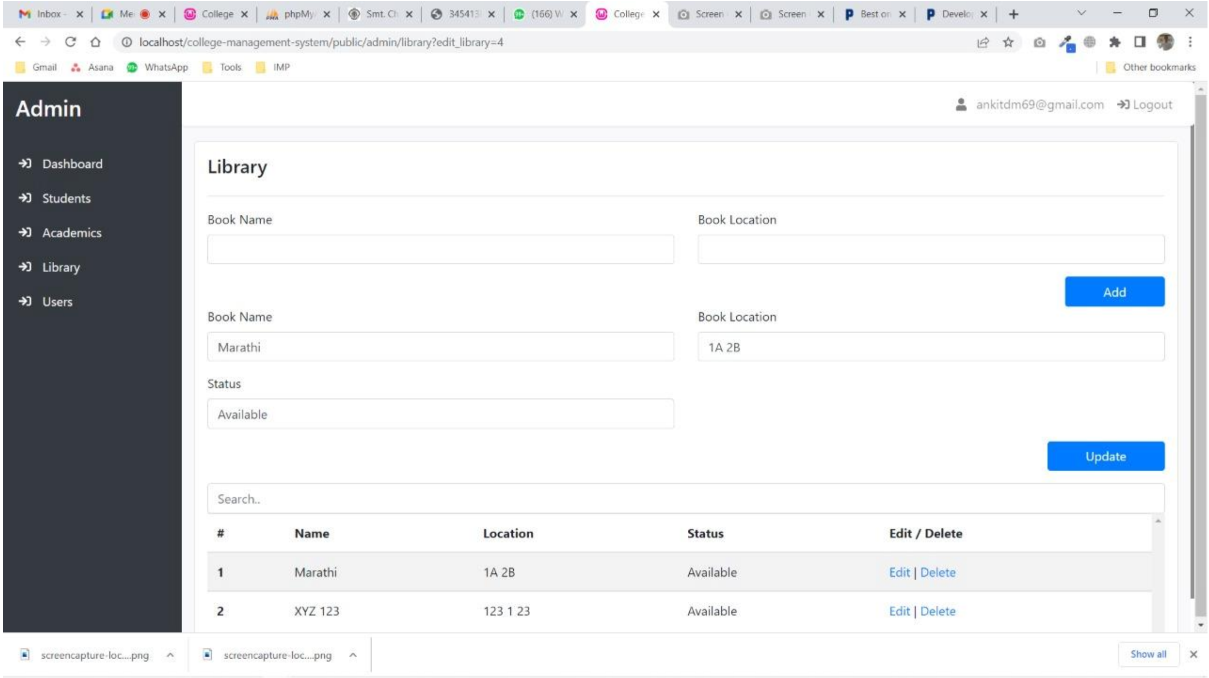
School Name
School Name

District
School District

State
School State

Country
School Country

Library



Library

Book Name:

Book Location:

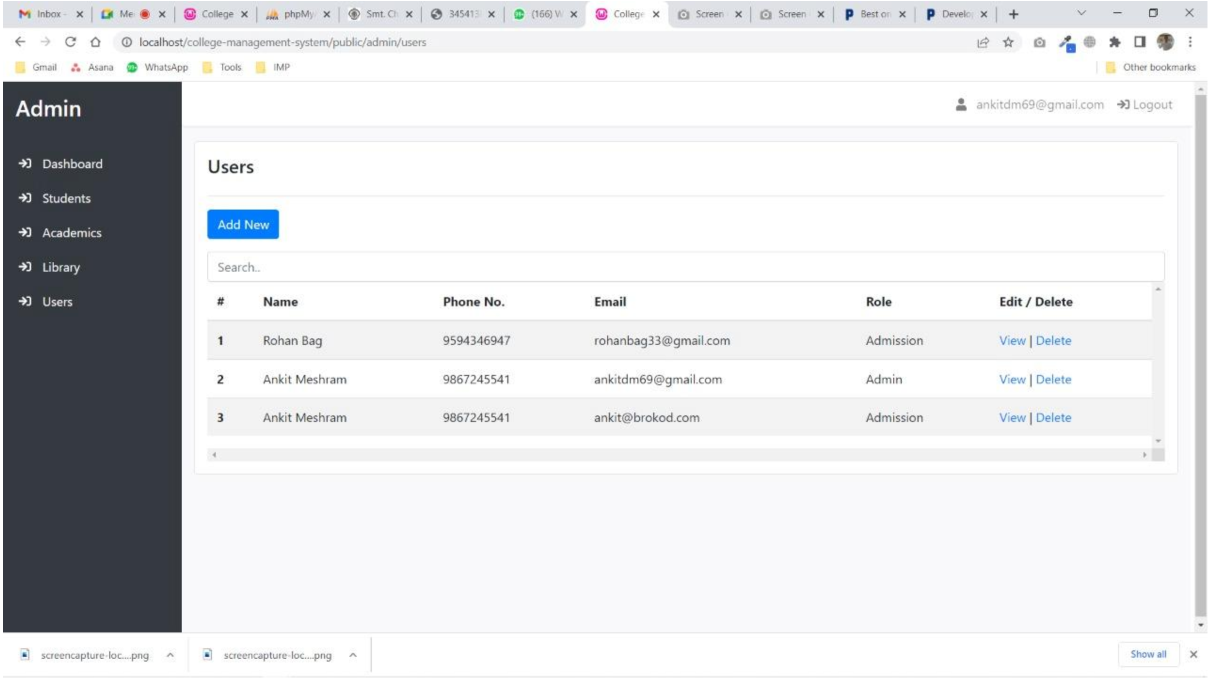
Book Name: Marathi

Book Location: 1A 2B

Status: Available

#	Name	Location	Status	Edit / Delete
1	Marathi	1A 2B	Available	Edit | Delete
2	XYZ 123	123 1 23	Available	Edit | Delete

Users maneger



Users

#	Name	Phone No.	Email	Role	Edit / Delete
1	Rohan Bag	9594346947	rohanbag33@gmail.com	Admission	View | Delete
2	Ankit Meshram	9867245541	ankitdm69@gmail.com	Admin	View | Delete
3	Ankit Meshram	9867245541	ankit@brokod.com	Admission	View | Delete

Chapter 7
CONCLUSIONS

7.1 Conclusion

The result of this project was a successfulll high-fidelity prototype of College management system (website). I successfully implemented all of the above mentioned use cases, and achieved clean interfaces that further enhance the user experience. The many-to-one database tables successfully represent a robust set of scenarios and relationships that are critical for a Tour Travel website. Admin can not only Add/Remove packagesCollege management system is to automate all functionalities of a college or university. Using this system you can manage all college management work like admission, fees submission, time table management and result declaration. Using this college management system you can view or update data and information about students and staff easily. This system helps in managing the activity like student admission, student registration, fees submission. Admin can also retrieve information of employee student. This College management system is designed to be user and administrator friendly, allowing both users and administrators to use it effectively.

7.2 Limitation of the system

Though we have put our best efforts to make the project/software to be altered in circumstances, easy to use but limitations cannot be controlled out by us. Some complicated or detailed options could not covered into to some extent because of logistics and experience it although software gives a wide range of options to the users. To make the software easy to use significant efforts we have put for those who does not relate to the fields of computer at the start layman find it difficult to use. The user is guided at each step of being able to proceed with the software without facing difficulty.

List of Limitations:

- Student attendance
- Lms (learning management system)
- Payroll management
- Attendance and leave
- Finance
- Hostel
- Visitor management

7.3 Future Scope of the Project

We have a plan to enhance this project in a very short time. Especially, we would like to implement the payment gateway for (visa card, credit card ctc). Moreover, We will also added some extend feature like Laundry facility, swimming pool facilities so the customer can fully enjoy the tour.

- Expanding the project in different location around the globe.
- Keep on changing UI – Design(User Interface) to make it look attractive.
- Making Servers more powerful.
- Expanding the databases for storing more customers information etc.
- Transforming this to the mobile application for better portability and making it more reliable.

- Student attendance
- Lms (learning management system)
- Payroll management
- Attendance and leave
- Finance
- Hostel
- Visitor management

References:

Text book reference

- Web design with HTML & CSS by Prem Kumar
- Javascript: The Good parts by Douglas Crockford
- PHP & MySQL by Laura Thomson and luke Welling
- Learning PHP, MySQL, javascript, CSS & HTML5: A step-by-step Guide to creating Dynamic Websites by Robin Nixon

➢ Link references:

- https://dotnet.microsoft.com/apps/aspnet
- https://www.tutorialspoint.com/
- http://www.geeks.com/
- http://youtube.com