HOSTEL MANAGEMENT SYSTEM

(ADMIN & HOD)

SUBMITTED IN PARTIAL FULFILLMENT OF

REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

OF

UNIVERSITY OF CALICUT

SUBMITTED BY

Ms. MUBASHIRA P

Reg. No: CUAVCMF018

UNDER THE GUIDANCE OF

Dr. LAJISH V L

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF CALICUT
2023

CERTIFICATE

This is to certify that the project work entitled “HOSTEL MANAGEMENT SYSTEM (ADMIN & HOD)” is a bonafide record of original work done by **Ms. MUBASHIRA.P** (CUAUCMF006) final year M.Sc. Computer Science in partial fulfillment of the requirements for the award of the degree in Master of Computer Science during the period of 2021-2023.

Project Guide:

Dr. LAJISH V L

Head of the Department

Certified of the candidate MUBASHIRA P (CUAVCMF018) is examined by us in the project viva voice.

Examination held on: …………………….

Signature of Examiners

1 . ……………………..

2 . ……………………..

Calicut university

Date:

DECLARATION

I here by declare that the project entitled **"HOSTEL MANAGEMENT SYSTEM (ADMIN & HOD)"** submitted to the **Department of Computer Science, University of Calicut** in partial fulfilment of the requirements for the award of degree in **MSc. COMPUTER SCIENCE** is a record of original dissertation work done by me, Under the guidance and supervision of **Dr.LAJISH V L Associate professor, Department of Computer Science, University of Calicut,Thenhipalam** during my study period in **CALICUT UNIVERSITY CAMPUS, MALAPPURAM.**

Place: Thenhipalam

MUBASHIRA P

Date:

Reg.No: CUAVCMF018

ACKNOWLEDGEMENT

Presentation, inspiration and motivation have always played a key role in the success of any venture.

I express my sincere thanks to my guide **Dr. LAJISH V L Associate Professor & Head of The Department, Department of Computer Science**, **University of Calicut**. Whose valuable guidance and kind supervision given to me through the course which shaped the present work as its show

I pay my deep sense of gratitude to **Ms. MANJULA K A Associate Professor ,Department of Computer Science, University of Calicut to** encourage me to the highest peak to provide me the opportunity to prepare project.

I wish to express my sincere thanks to faculty **Dr. REMYA CHANDRAN Assistant Professor, Ms. ALJINU KHADAR K V Assistant Professor, Department of Computer Science, University of Calicut** for their valuable suggestion and guidance throughout our project that helped us to develop and complete the project successfully.

I'm immensely obliged to my friends for their elevating inspiring, encouraging guidance and kind of supervision in the completion of my project.

ABSTRACT

The project HOSTEL MANAGEMENT SYSTEM is an integrated management system which will help to reduce the errors mostly caused by human mistakes and also eliminate the time delay and issues of using file records. The drawbacks of existing manual system lead to designing of a computerized system that will be compatible to provide better services to students and staffs. The goal of this web application is to develop a solution to transfer the current manual practices into an automated system. It is provided with a better control over entire hostel system like student registration, room allocation and deallocation, tracking mess details and finance section.

HOSTEL MANAGEMENT SYSTEM is software developed for managing various activities in the hostel. For the past few years, the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution, and hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

INDEX

INTRODUCTION

INTRODUCTION

In our current era of automated systems with it being either software or hardware, it’s not advisable to be using manual system. Hostels without a management system are usually done manually.Registration forms verification to other data saving processes are done manually and most at times, they are written on paper. Thus a lot of repetitions can be avoided with an automated system. The drawbacks of existing systems lead to the design of a computerized system that will help reduce a lot of manual inputs. With this system in place, we can improve the efficiency of the system, thus overcome the drawbacks of the existing manual system. This system is designed in favour of the hostel management which helps them to save the records of the students about their rooms and other things. It helps them from the manual work from which it is very difficult to find the record of the students and the mess bills of the students, and the information of about those ones who had left the hostel years before.

Objectives

The main objective of the Hostel Management System is to manage the details of Students,Allocation, Mess, Rooms, Rent, Payments and Deallocation. It manages all the information about Students, Rooms, Payments and Mess. The purpose of the project is to build an application program to reduce the manual work for managing the Students, Allocation, Mess, Rooms, Rent, Payments and Deallocation. It tracks all the details about the Hostel, Rooms, Payments.

- Efficiently allocate all students and store their details automatically after allocation.
- Hod can verify student and give approval for allocation.
- Admin can add and edit notice board,departments,courses etc.
- Room admin can allocate and deallocate students.
- Student can add complaint and also can view reply of the complaint.
- Mess and Room admin can view feedbacks provided by the students.
- Room admin add student details after allocation and it deletes the details after deallocation

SYSTEM STUDY

SYSTEM STUDY

System study is the first stage of system development life cycle. This gives a clear picture of what actually the physical system is. The system study is done in two phases. In the first phase, the preliminary survey of the system is done which helps in identifying the scope of the system. The second phase of the system study is more detailed and in-depth study in which the identification of user's requirement and the limitations and problems of the present system are studied. After completing the system study, a system proposal is prepared by the user. Actually, in this project I just gather information to public, some social service centers and so on. After the preliminary study I got some knowledge about current tastes of Customers. And also what are the various types of products, how to give product quality of each item and current marketing value of each product and so on.

EXISTING SYSTEM

The Existing system is completely manual and need lot of efforts and consume enough time. All the hostel records are to be maintained for the details of each student, fee details, room allocation, mess details, etc. All these details are entered and retrieved manually. It may lead to corruptions in the allocation process as well as hostel fee calculation. The existing system does not deals with mess calculation and complaint registration.

Drawbacks:

- More human power
- More strength and strain of manual labour needed
- Repetition of same procedure.
- Low security.
- Data redundancy.
- Difficulty to handle.
- Difficulty to update data.
- Record keeping is difficult.
- Backup data can be easily generated.

PROPOSED SYSTEM

The proposed system is having many advantages over the existing system. It requires less overhead and very efficient. This project is aimed at developing a system for keeping records and showing information about or in a hostel. This system will help the hostel officer to be able to manage the affairs of the hostel. This system will provide full information about a student in the hostel. It will show rooms available or not and number of people in a particular room. This will also provide information on students who have paid in full or are still owing. This system will also provide a report on the summary detail regarding fees and bills students are owing. The proposed system deals with the mess calculation and allotment process efficiently.

Advantages:

- User friendly
- Easier and flexible
- Centralized database
- Good security for user information

MODULE DESCRIPTION

The project HOSTEL MANAGEMENT SYSTEM is an integrated management system which will help to reduce the errors mostly caused by human mistakes and also eliminate the time delay and issues of using file records. The drawbacks of existing manual system leads to designing of a computerized system that will be compatible to provide better services to students and staffs. The goal of this web application is to develop a solution to transfer the current manual practices into an automated system. It is provided with a better control over entire hostel system like student registration, room allocation and deallocation, tracking mess details and finance section. This particular project deals with problem on managing a Hostel and avoids the problem which occur when carried manually.

- Admin Module
- Hod Module
- Student Module
- Mess Module
- Room Module

ADMIN MODULE

In administrator module administrator manages the master data like server details and student details. Add and approve the admins and hod, can add,edit and delete departments, course, add notifications, view the students in the hostel,view transient message details,generate and view admission fees and category etc.

HOD MODULE

In Hod module Hod manages the details of student registration. Hod verifies the students and provide approve to the room admin for further room allocation.Hod can add his/her student details .Hod can change position transiently.

SOFTWARE AND HARDWARE SPECIFICATION

HARDWARE SPECIFICATION

System	: Intel
Processor	: Core i3
Ram Capacity	: 4GB
Hard Disk Drive	: 1TB
Keyboard	: Standard
Mouse	: Standard

SOFTWARE SPECIFICATION

Platform	: Windows OS
Front End	: Python, Flutter, HTML, CSS
Back End	: Django
API	: Django
Browser	: Google Chrome, Mozilla Firefox, Internet Explorer

LANGUAGE DESCRIPTION

HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup languages are human readable.

HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991. The first ever version of HTML was HTML 1.0.HTML uses predefined tags and elements which tells the browser about content display property. If a tag is not closed then browser applies that effect till end of page.

CSS (Cascading Style Sheet)

A CSS (cascading style sheet) file allows you to separate your web sites HTML content from its style. As always you use your HTML file to arrange the content, but all of the presentation (fonts, colors, background, borders, text formatting, link effects & so on...) are accomplished within a CSS. At this point you have some choices of how to use the CSS, either internally or externally. First, we will explore the internal method. This way you are simply placing the CSS code within the tags of each HTML file you want to style with the CSS. The format for this is shown in the example below. Inline styles are defined right in the HTML file along side the element you want to style. An external CSS file can be created with any text or HTML editor. A CSS file contains no (X)HTML, only CSS. You simply save it with the .css file extension.

FLUTTER

Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile application in Google's own language, *Dart*. This tutorial walks through the basics of Flutter framework, installation of Flutter SDK, setting up Android Studio to develop Flutter based application, architecture of Flutter framework and developing all type of mobile applications using Flutter framework. Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML.to be specific, Flutter application is itself a widget.

DJANGO

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLlite3, etc. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use and that too for rapid development. Django's primary goal is to ease the creation of complex database-driven websites. Some well known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket and Mozilla.

PYTHON

Python is a general-purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.Python is an easy-to-learn yet powerful and versatile scripting language, which makes it attractive for Application Development. With its interpreted nature, Python's syntax and dynamic typing make it an ideal language for scripting and rapid application development. Python supports multiple programming patterns, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. It is a multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.We don't need to use data types to declare variable because it is dynamically typed, so we can write a=10 to assign an integer value in an integer variable. Python makes development and debugging fast because no compilation step is included in Python development, and the edit-test-debug cycle is very fast. Python has many web-based assets, open-source projects, and a vibrant community. Learning the language, working together on projects, and contributing to the Python ecosystem are all made very easy for developers. Because of its straightforward language framework, Python is easier to understand and write code in. This makes it a fantastic programming language for novices. Additionally, it assists seasoned programmers in writing clearer, error-free code.

SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is concerned with analyzing, designing, implementing and evaluating information system in our organization. It is carried out to make the system more effective either by modification or by substantial redesign. In system analysis we identify the problem, study the alternative solution and select the most suitable solution, which meet the technical economic and social demands for analysis, various tools such as dataflow diagram, interviews on site observation, questionnaires etc., are used. System analysis process is also called a life cycle methodology since it relates to four significant phases in life cycle of all information system. They are

1. System Analysis / Study Phase.
2. System Design / Design phase.
3. System Development / Development Phase.
4. Testing and implementation / Operation Phase.

All activities associated with each life cycle phase must be performed managed and documented. So, system analysis is the performance, management and documentation of the activities related to the four life cycle phases of a computer-based system

.

FEASIBILITY STUDY

The most difficult part of feasibility analysis is the identification of the Candidate system and the evaluation of their performance. Feasibility study is a test of a system proposal according to its workability, impact on the organization, ability to meet the user needs and effective use of resources.

A feasibility study is conducted to select the best system that meets performance requirements. The entails an identification description, an evaluation of candidate system, and the selection of the best system for the job. The new system has advantages such as we can easily doing transactions in the shop and this application is more user friendly for the employees. Six key considerations are involved in the feasibility analysis:

1. Technical Feasibility
2. Operational Feasibility
3. Economical Feasibility
4. Behavioral Feasibility
5. Software Feasibility
6. Hardware Feasibility

TECHNICAL FEASIBILITY

A study of function, performance and constraints may improve the ability to create an acceptable system. Technical Feasibility is frequently the most difficult area to achieve at the stage of product engineering process. Considering that are normally associated with the technical feasibility include,

- Development risk
- Resource availability
- Technology

Technical Feasibility study deals with the hardware as well as software requirements. The scope was whether the work for the project is done with the current requirements and existing software technology has to be examined in the feasibility study.

The outcome was found to be positive. In the proposed system, data can be easily stored and managed using database management system software. The reports and results for various queries can be generated easily. Therefore, the system is technically feasible.

OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned into information system that will meet the organization's operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are these major barriers to implementation?

The purpose of the operational feasibility study is to determine whether the new system will be used if it is developed and implemented from users that will undermine the possible application benefits. There was no difficulty in, implementing the system and the proposed system is so effective, user friendly and functionally reliable so that the users in the company will find that the new system reduce their hard steps. If the user of the system is fully aware of the internal working of the system then the users will not be facing any problem in running the system.

ECONOMICAL FEASIBILITY

Proposed system was developed with the available resources. Since cost input for the software is almost nil the output of the software is always a profit. Hence software is economically feasible. In the existing system, manpower is more required. In the proposed system the effort to be involved is reduced drastically. So, the proposed system is said to be economic.

BEHAVIORAL FEASIBILITY

People are inherently resistant to changes and computer is known as facilitating the changes. An estimate should be made of how strongly the users react to the development of the system. The proposed system consumes time. Thus the people are made to engage in some other important work.

SOFTWARE FEASIBILITY

Even though software is developed in very high software environment, it will be supported by many other and environments with minimum changes.

HARDWARE FEASIBILITY

The software can be developed with resource already existing. Here the consideration is that the existing hardware resources support the technologies that are to be used by the new system. No hardware was newly bought for the project and hence software is said to achieve hardware feasibility. The software can be developed with resource already existing. Here the consideration is that the existing hardware resources support the technologies that are to be used by the new system. No hardware was newly bought for the project and hence software is said to achieve hardware feasibility.

SYSTEM DESIGN

SYSTEM DESIGN

System Design involves translating system requirements and conceptual design into technical specification and general flow of processing. After the system requirements have been identified, information has been gathered to verify the problems and after evaluating the existing system a new system is proposed. System Design is the process of planning of new system or to replace or complement an existing system. It must be thoroughly understood about the old system determine how computers can be used to make its operations more effective.

System Design sits at technical the kernel of the system development. Once system requirements have been analyzed and specified system design is the first of the technical activities – design, code generation and test that required to build and verify the software. System Design is the most creative and challenging phases of the system life cycle. The term design describes the final system and the process by which it is to be developed.

System Design is the high-level strategy for solving the problem and building a solution. System Design includes decisions about the organization of the system into subsystems, the allocation of subsystems to hardware and software components and major conceptual and policy decision that forms the framework for detailed design.

DATA FLOW DIAGRAM

A Data Flow Diagram is used to define the flow of data and the resources such as information. Data Flow Diagrams are a way of expressing system requirements in graphical manner .it has the purpose of clarifying system requirements and identifying the major transformation that will become program in the system design. So it is the starting point of design phase that functionally decomposes the requirement specification down in to the lowest level of details. The bubbles represent data transformation and the lines represent information flow in the system. Data Flow Diagrams are useful in understanding a system and can be effectively used for partitioning. The system may be an organization, a manual procedure, software system, a mechanical system or any combination of these.

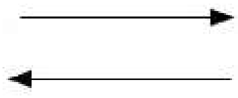
Rules For Constructing a Data Flow Diagram

Process should be named and numbered for easy reference. Each name should be Representative of process. The direction of flow is from top to bottom and from left to right. That is information flow should be from source to destination. Numbering is given when a process is exploded into lower-level details. The name of the dad stores, source and destination are written in capital letters. Process and Data Flow names have the first letter of each word capitalized. The Data Flow Diagram is particularly designed to aid communication. If it contains dozens of process and data stores it gets too unwieldy. The rule of the thumb is to explode the DFD into a functional level beyond that, it is best to take each function separately and expand it to show the explosion in a single process. If a user wants to know what happens within a given process, then the detailed explosion of that process may be shown.

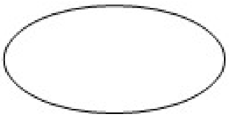
The goal of DFD is to have a commonly understood model of a system. The diagram is the basis of structured system analysis. DFD are supported by other techniques of structured system analysis. DFD are supported by other techniques of structured system analysis such as structured diagrams, and data dictionaries.

DFD SYMBOLS

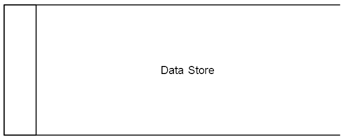
Data Flow Diagrams are composed of the four basis symbols shown below



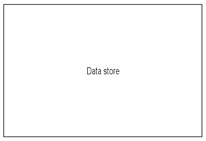
A data is a root, which enables packet of data to travel from one point to another. Data may flow from a source to a processor and from data source or process. An arrow line depicts the flow, with arrowhead pointing in the direction of flow.



A process represents transformation where incoming data changed into outgoing data flows.



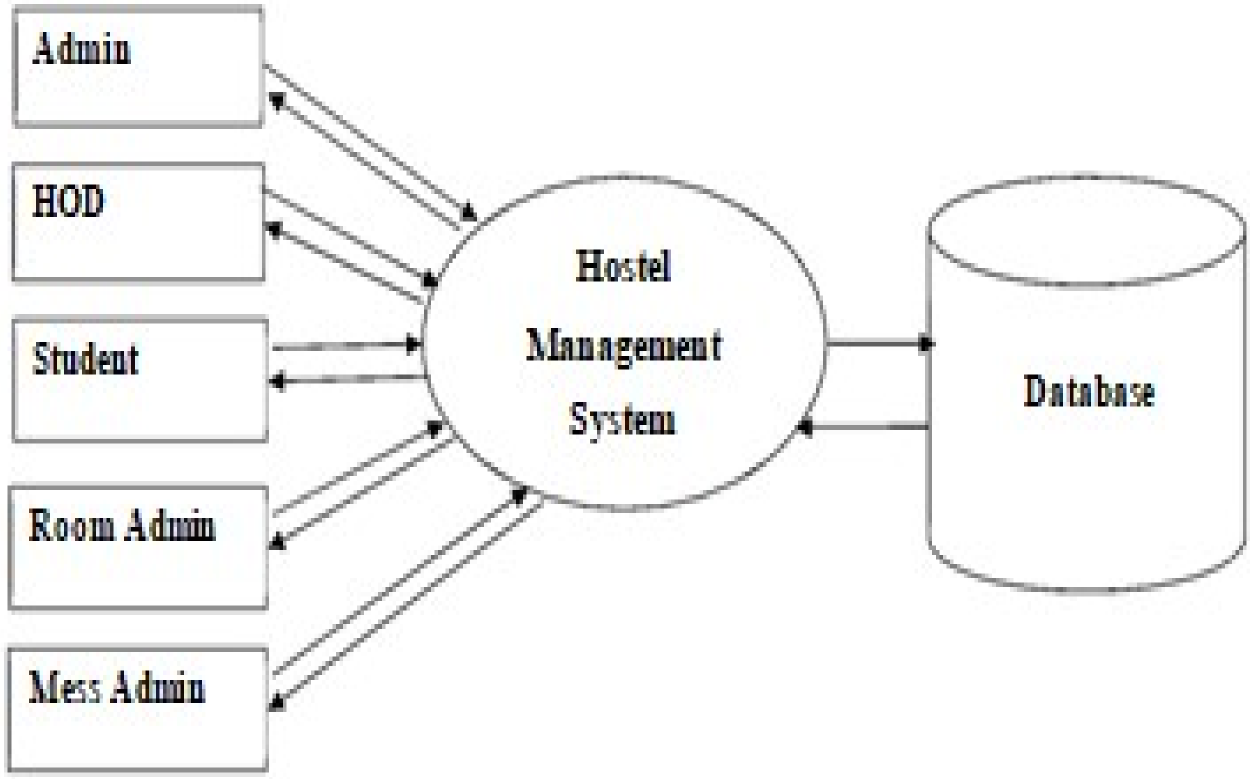
An open-ended box represents a data store, data at rest or a temporary repository of data.



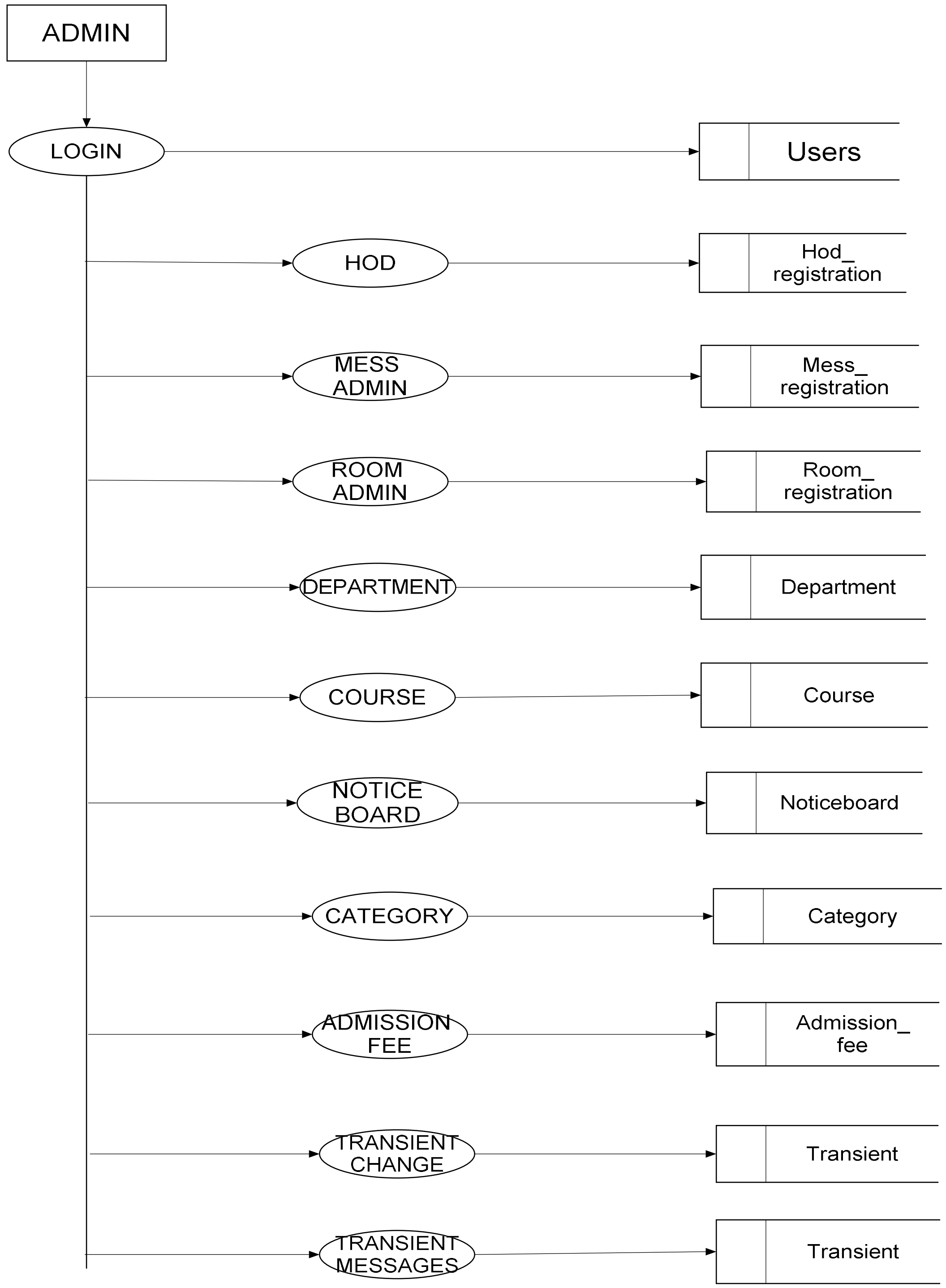
A square defines a source or destination of system data.

LEVEL 0

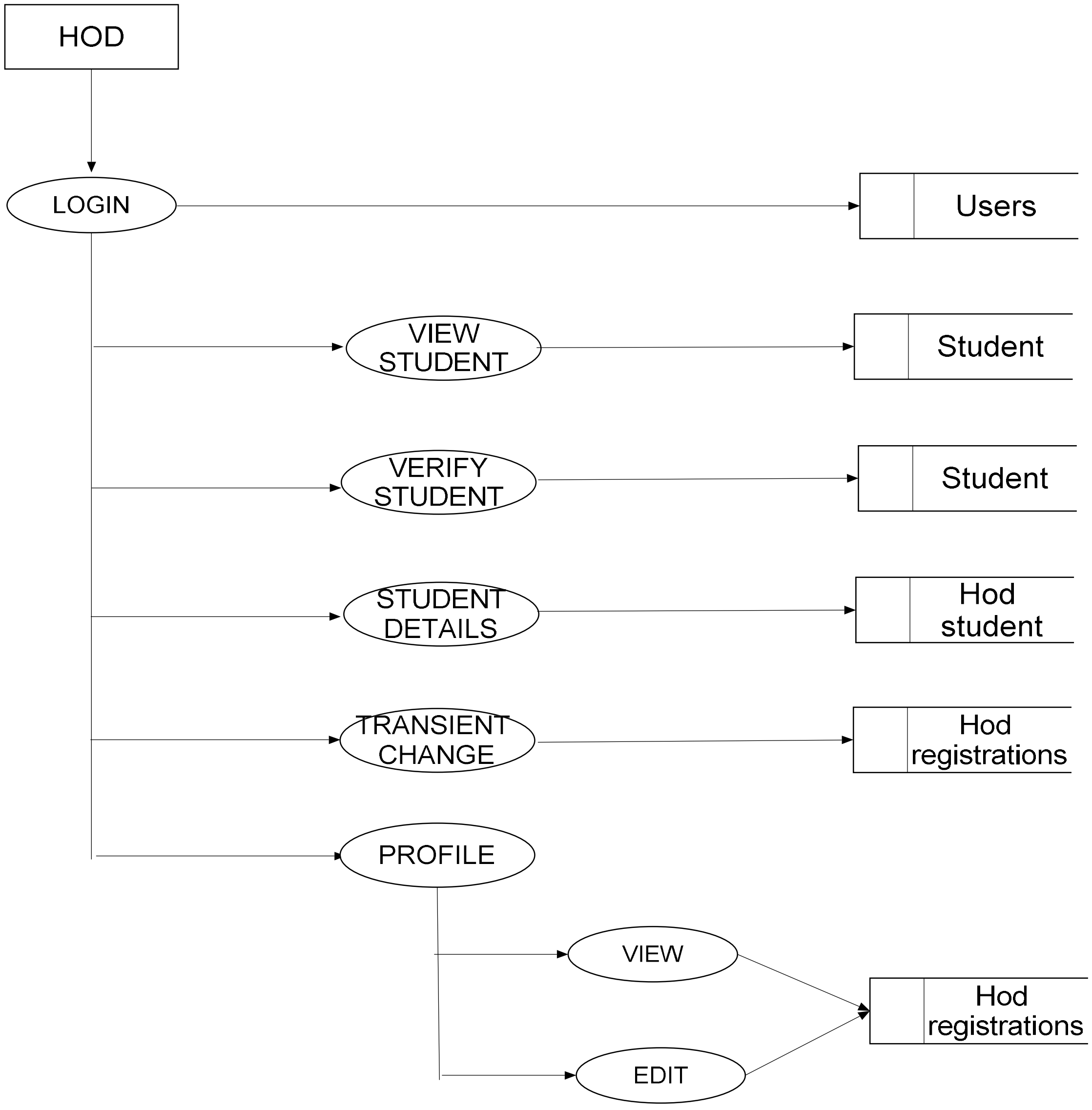
Data Flow Diagrams are composed of the four basis symbols shown below



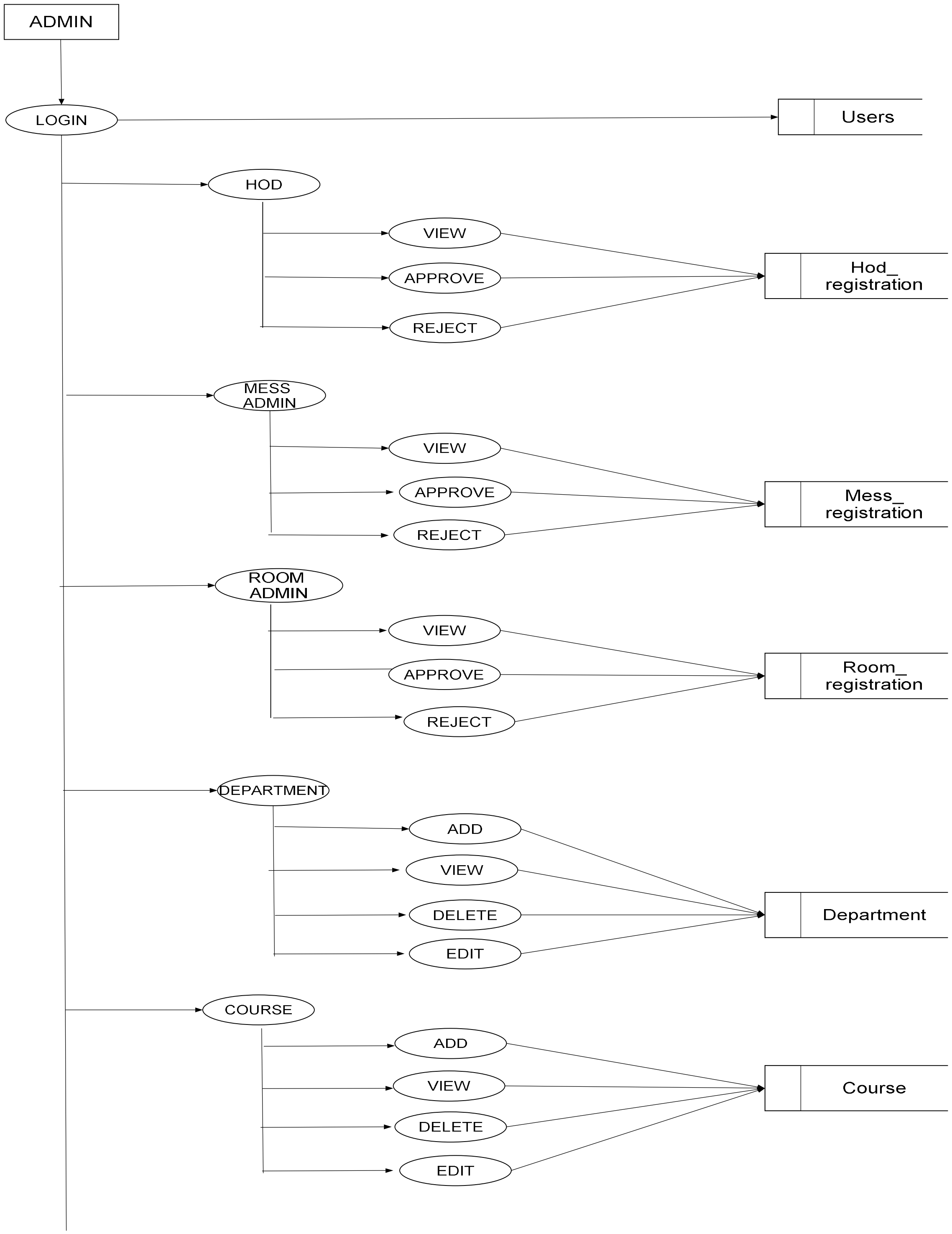
LEVEL 1: ADMIN

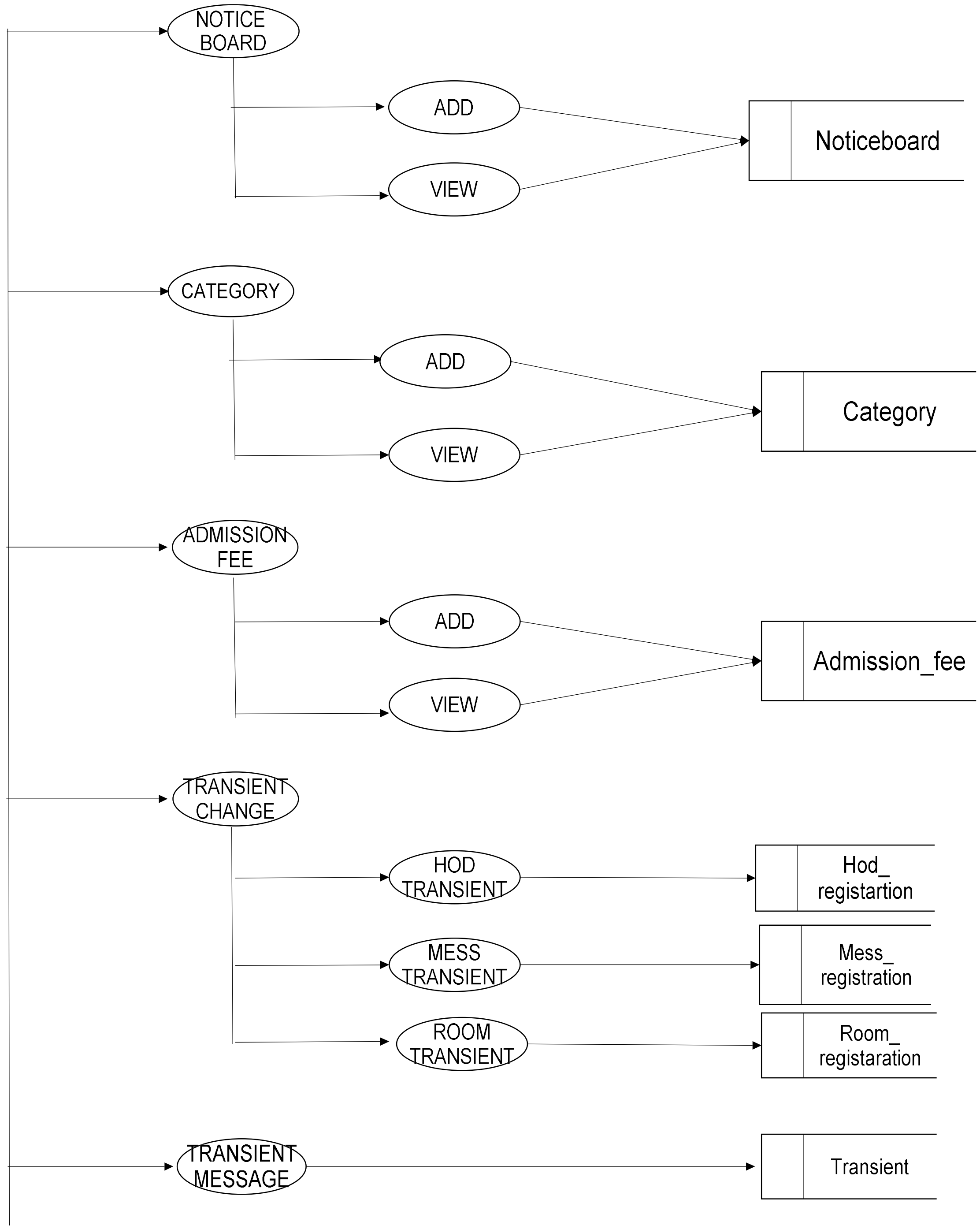


LEVEL 1: HOD


HOD
LOGIN
Users
VIEW STUDENT
Student
VERIFY STUDENT
Student
STUDENT DETAILS
Hod student
TRANSIENT CHANGE
Hod registrations
PROFILE
VIEW
EDIT
Hod registrations

LEVEL 2: Admin



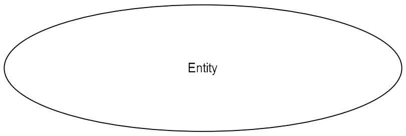

NOTICE BOARD
ADD
VIEW
Noticeboard
CATEGORY
ADD
VIEW
Category
ADMISSION FEE
ADD
VIEW
Admission_fee
TRANSIENT CHANGE
HOD TRANSIENT
Hod_ registartion
MESS TRANSIENT
Mess_ registration
ROOM TRANSIENT
Room_ registaration
TRANSIENT MESSAGE
Transient

ENTITY RELATIONSHIP DIAGRAM

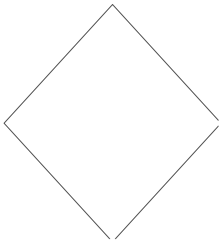
An E R diagram is a model that identifies the concept or entities that exist in a system and the relationship between those entities. An ERD is often used as a way to visualize a relational database each entity represents a database table and the relationship lines represents the key in one table that point to specific records in related tables.

- Entity :-

- Attribute :-



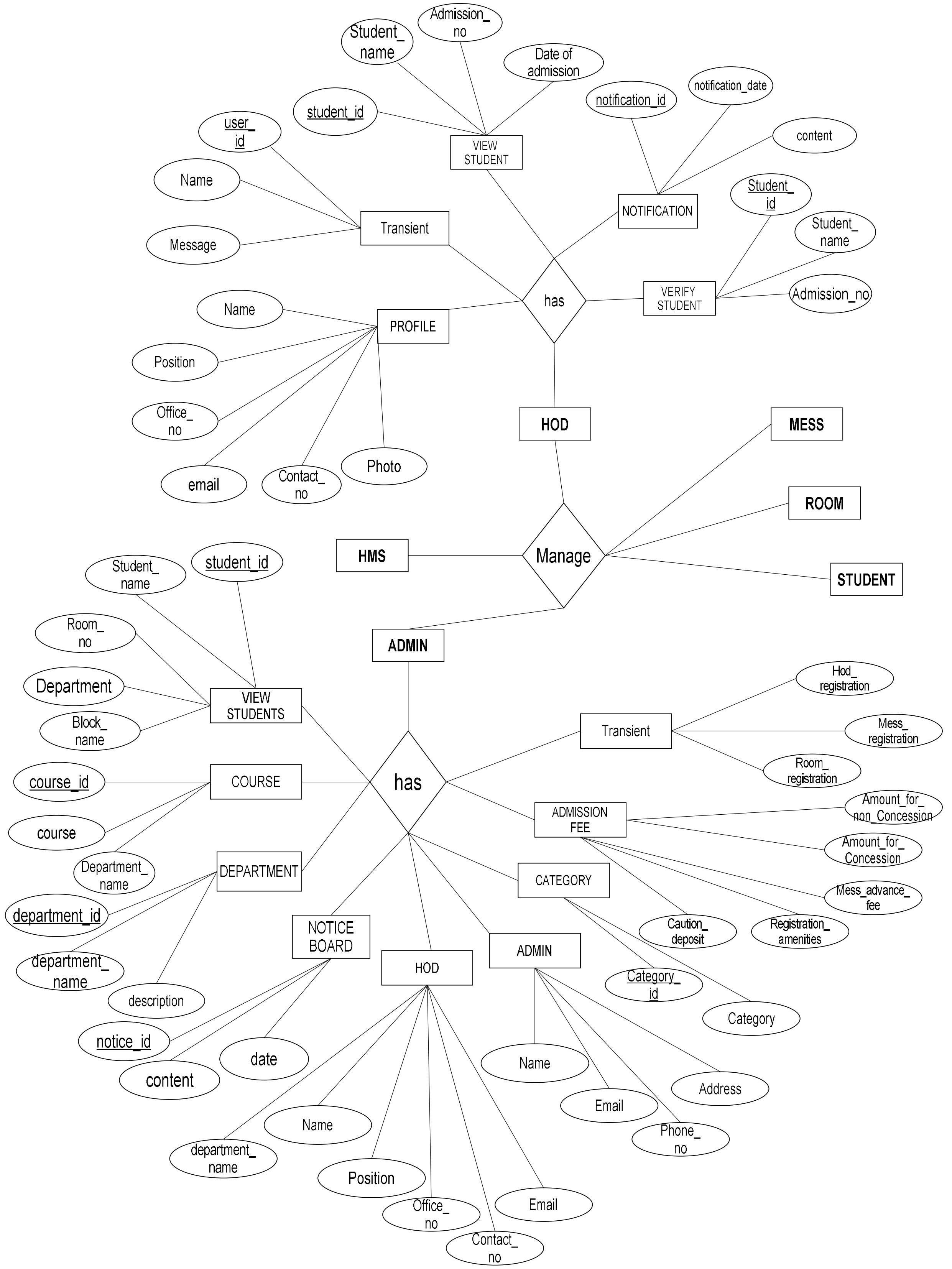
- Relationship :-



It depicts the fundamental relations like recording personal information, paying salary and getting a loan. ER involved the student information and payroll system.

Advantage of ER diagram

- Professional and faster development.
- Productivity improved.
- Fewer faults in development.
- Maintenance becomes easy.


Student_ name
Admission_ no
Date of admission
student_id
VIEW STUDENT
notification_id
notification_date
content
NOTIFICATION
user_ id
Name
Message
Transient
Student_ id
Student_ name
Admission_no
VERIFY STUDENT
has
Name
PROFILE
Position
Office_ no
email
Contact_ no
Photo
HOD
MESS
ROOM
HMS
Manage
STUDENT
ADMIN
Student_ name
student_id
Room_ no
Department
VIEW STUDENTS
Block_ name
Hod_ registration
Transient
Mess_ registration
Room_ registration
has
course_id
COURSE
course
Department_ name
ADMISSION FEE
Amount_for_ non_Concession
Amount_for_ Concession
Mess_advance_ fee
DEPARTMENT
department_id
department_ name
description
CATEGORY
Caution_ deposit
Registration_ amenities
NOTICE BOARD
HOD
ADMIN
Category_ id
Category
notice_id
content
date
Name
Address
Email
Phone_ no
Name
department_ name
Position
Office_ no
Contact_ no
Email

DATABASE DESIGN

The data base design is a logical development in the methods used by the computer to access and manipulate data stored in the various parts of the computer system. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as organization recourses and as an integrated while. The main objectives of database are data integration, data integrity and data independence.

SQLITE

SQLITE is an in-process library that implements a self-contained, serverless, zero- configuration, Transactional SQL database engine. It is a database, which is zero configured, which means like Other database you do not need to configure it in your system. SQLITE engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLITE accesses its storage files directly. a complete SQLITE database is stored in a single cross platform disk file. Its transactions are fully ACID-complaint, allowing safe access from multiple processes or threads. It is written in ANSI-C and provide easy to use API.
SQLITE available on UNIX , LINUX, Mac OS-X, and Windows

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. Redundant data wastes disk space and creates maintenance problems. If data that exists in more than one place must be changed, the data must be changed in exactly the same way in all locations. Database normalization is a database schema design technique, by which an existing schema is modified to minimize redundancy and dependency of data.

1 st Normal Form (1NF)

In this Normal Form, we tackle the problem of atomicity. Here atomicity means values in the table should not be further divided. In simple terms, a single cell cannot hold multiple values. If a table contains a composite or multi-valued attribute, it violates the First Normal Form.

2 nd Normal Form (2NF)

The first condition in the 2nd NF is that the table has to be in 1st NF. The table also should not contain partial dependency. Here partial dependency means the proper subset of candidate key determines a non-prime attribute.

3 rd Normal Form (3NF)

The same rule applies as before i.e., the table has to be in 2NF before proceeding to 3NF. The other condition is there should be no transitive dependency for non-prime attributes. That means non-prime attributes (which doesn't form a candidate key) should not be dependent on other non-prime attributes in a given table. So a transitive dependency is a functional dependency in which X → Z (X determines Z) indirectly, by virtue of X → Y and Y → Z (where it is not the case that Y → X)

TABLE DESIGN

1. **Table Name: Login**

Sl.No	Field	Data type	Constraints
1	Login_id	Integer Field	Primary Key
2	Role	Char Field(50)	
3	Username	Char Field(50)	
4	Password	Char Field(50)	

2. **Table Name: Noticeboard**

Sl.No	Field	Data type	Constraints
1	Noticeboard_id	Integer Field	Primary Key
2	User_type_id	Integer Field	Foreign Key
3	Title	Char Field(100)	
4	Content	Char Field(500)	
5	Date	Date Field()	

3. **Table Name: Department**

Sl.No	Field	Data type	Constraints
1	Department_id	Integer Field	Primary Key
2	Department_name	Char Field(100)	
3	Department_description	Char Field(500)	

4. **Table Name: Course**

Sl.No	Field	Data type	Constraints
1	Course_id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign key
3	Department_name	Char Field(100)	
4	Course_name	Char Field(100)	

5. **Table Name: User_type**

Sl.No	Field	Data type	Constraints
1	User_type_id	Integer Field	Primary Key
2	Type	Char Field(50)	

6. **Table Name: Room_type**

Sl.No	Field	Data type	Constraints
1	Room_type_id	Integer Field	Primary Key
2	Room_type	Char field(200)	

7. **Table Name : Category**

Sl.No	Field	Data type
1	Category _id	Integer Field
2	Category	Char field(200)

8. **Table Name: Room**

Sl.No	Field	Data type	Constraints
1	Room_id	Integer Field()	Primary Key
2	Block_id	Integer Field()	Foreign Key
3	Room_type_id	Integer Field()	Foreign Key
4	Max_accommodation	Integer Field()	
5	Vaccancy	Integer Field()	
6	Block_name	Char Field(100)	
7	Rooms_type	Char Field(100)	
8	Amount	Integer Field()	
9	Room_no	Char Field(100)	

9. **Table Name: Block**

Sl.No	Field	Data type	Constraints
1	Block_id	Integer Field()	Primary Key
2	Block_name	Char Field(100)	
3	Block_description	Char Field(200)	

10. Table Name: Room_rent

Sl.No	Field	Data type	Constraints
1	Room_rent_id	Integer Field()	Primary Key
2	Department_id	Integer Field()	Foreign Key
3	Student _id	Integer Field()	Foreign Key
4	Block_id	Integer Field()	Foreign Key
5	Room_id	Integer Field()	Foreign Key
6	Payment_status	Char Field(100)	
7	Department_name	Char Field(100)	
8	Rent	Integer Field()	
9	Room_no	Char Field(100)	
10	Student_name	Char Field(100)	
11	Block_name	Char Field(100)	

11. Table Name: Hod_transient

Sl.No	Field	Data type	Constraints
1	Hod_student_id	Integer Field()	Primary Key
2	Student_name	Char Field(200)	
3	Join_date	Date Field()	
4	Admisson_no	Integer Field()	

12. Table Name: Hod_registration

Sl.No	Field	Data type	Data type
1	Hod_registration_id	Integer Field	Primary Key
2	User_type_id	Integer Field	Foreign Key
3	Department_id	Integer Field	Foreign Key
4	Name	Char Field(200)	
5	Position	Char Field(200)	
6	Email	Char Field(200)	
7	Office_no	Integer Field()	
8	Contact_no	Integer Field()	
9	Photo	File Field()	
10	Username	Char Field(200)	
11	Password1	Char Field(200)	
12	Password2	Char Field(200)	
13	Department_name	Char Field(200)	
14	Role	Char Field(200)	
15	Proof	File Field()	

13. Table Name: Hod_transient

Sl.No	Field	Data type	Constraints
1	Hod_student_id	Integer Field()	Primary Key
2	Student_name	Char Field(200)	
3	Join_date	Date Field()	
4	Admisson_no	Integer Field()	

14. Table Name: Admission_fee

Sl.No	Field	Data type	Constraints
1	Admission_fee_id	Integer Field()	Primary Key
2	Caution_deposit	Integer Field()	
3	Registration_amenities	Integer Field()	
4	Mess_advance_fee	Integer Field()	
5	Amount_for_concession	Integer Field()	
6	Amount_for_oec	Integer Field()	
7	Amount_for__non_concession	Integer Field()	

15. Table Name: Room_fee

Sl.No	Field	Data type	Data type
1	Room_fee_id	Integer Field	Primary Key
2	Month	Char Field(100)	
3	Amount	Integer Field	

16. Table Name: Mess_fee

Sl.No	Field	Data type	Data type
1	Mess_fee_id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Paid_date	Integer Field	Foreign Key
5	No_of_days_present	Date Field()	
6	Mess_fee	Char Field(500)	
7	Total	Char Field(200)	
8	Month_year	Char Field(200)	
9	Department_name	Char Field(200)	
10	Student_name	Char Field(200)	

17. Table Name: Complaint

Sl.No	Field	Data type	Data type
1	Complaint_id	Integer Field	Primary Key
2	Block_id	Integer Field	Foreign Key
3	Department_id	Integer Field	Foreign Key
4	Student_id	Integer Field	Foreign Key
5	Room_id	Integer Field	Foreign Key
6	Date	Date Field()	
7	Complaint_description	Char Field(500)	
8	Reply	Char Field(200)	
9	Title	Char Field(200)	
10	Department_name	Char Field(200)	
11	Block_name	Char Field(200)	
12	Student_name	Char Field(200)	
13	Room_no	Char Field(200)	

18. Table Name: Mess_complaint

Sl.No	Field	Data type	Data type
1	Complaint_id	Integer Field	Primary Key
2	Block_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Date	Date Field()	
5	Complaint_description	Char Field(500)	
6	Reply	Char Field(200)	
7	Title	Char Field(200)	
8	Block_name	Char Field(200)	
9	Student_name	Char Field(200)	

19. Table Name: Mess_registration

Sl.No	Field	Data type	Data type
1	Mess_registration_id	Integer Field	Primary Key
2	User_type_id	Integer Field	Foreign Key
3	Name	Char Field(200)	
4	Email	Char Field(200)	
5	DOB	Date Field	
6	Phone_no	Integer Field	
7	Address	Char Field(500)	
8	Photo	File Field()	
9	Proof	File Field()	
10	Username	Char Field(200)	
11	Password1	Char Field(200)	
12	Password2	Char Field(200)	
13	Role	Char Field(200)	

20. Table Name: Coupon

Sl.No	Field	Data type	Constraints
1	Coupon_id	Integer Field()	Primary Key
2	Type	Char Field(100)	
3	Amount	Integer Field()	

21. Table Name: Coupon_generate

Sl.No	Field	Data type	Constraints
1	Coupon_generate_id	Integer Field()	Primary Key
2	Coupon_id	Integer Field()	Foreign Key
3	Student_id	Integer Field()	Foreign Key
4	Issued_date	Date Field()	
5	Amount	Integer Field()	
6	Type	Char field(50)	
7	Student_name	Char field(50)	

22. Table Name: Mess_track

Sl.No	Field	Data type	Data type
1	Mess_track _id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Block_id	Integer Field	Foreign Key
5	Mess_in_date	Date Field	
6	Mess_out_date	Date Field	
7	In_or_out	Char Field(500)	
8	Time	Char Field(200)	
9	Department_name	Char Field(200)	
10	Student_name	Char Field(200)	
11	Date	Date Field	
12	Block_name	Char Field(200)	
13	Mess_in_count	Integer Field	
14	Mess_out_count	Integer Field	
15	Students_present	Integer Field	
16	Month	Char Field(200)	

23. Table Name: Room_registration

Sl.No	Field	Data type	Data type
1	Room_registration_id	Integer Field	Primary Key
2	User_type_id	Integer Field	Foreign Key
3	Name	Char Field(200)	
4	Email	Char Field(200)	
5	DOB	Date Field	
6	Phone_no	Integer Field	
7	Address	Char Field(500)	
8	Photo	File Field()	
9	Proof	File Field()	
10	Username	Char Field(200)	
11	Password1	Char Field(200)	
12	Password2	Char Field(200)	
13	Role	Char Field(200)	

24. Table Name: Expenditure

Sl.No	Field	Data type	Data type
1	Expenditure_id	Integer Field	Primary Key
2	Guest_Coupon_id	Integer Field	Foreign Key
3	Year	Integer Field	
4	Month	Integer Field	
5	Total_purchase	Float Field	
6	Opening_stock	Float Field	
7	Closing_stock	Float Field	
8	Net_purchase	Float Field	
9	Guest_coupon_charge	Integer Field	
10	Total_mess_expenditure	Integer Field	
11	Pay_per_expense	Float Field	
12	Salary	Float Field	
13	Pf	Float Field	
14	Mis	Float Field	
15	Audit_fee	Float Field	
16	Printing_stationary	Float Field	
17	Other_expense	Float Field	
18	Total_fixed_expenditure	Float Field	
19	No_of_inmates	Float Field	
20	Expense_per_inmates	Float Field	
21	Total_mess_attendance	Float Field	

25. Table Name: Room_request

Sl.No	Field	Data type	Data type
1	Room_request _id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Date	Date Field()	
5	Verification_status	Char Field(200)	
6	Distance_from_home	Char Field(500)	
7	Admission_no	Integer Field	
8	Department_name	Char Field(200)	
9	Student_name	Char Field(200)	

26. Table Name: Deallocation_request

Sl.No	Field	Data type	Data type
1	Dealloction_request _id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Block_id	Integer Field	Foreign Key
5	Room_id	Integer Field	Foreign Key
6	Reason	Char Field(200)	
7	Date	Date Field	
8	Block_name	Char Field(200)	
9	Student_name	Char Field(200)	
10	Room_no	Char Field(200)	
11	Department_name	Char Field(200)	
12	Payment_status	Char Field(200)	

27. Table Name: Student

Sl.No	Field	Data type	Data type
1	Student_id	Integer Field	Primary Key
2	Log_id	Integer Field	Foreign Key
3	Department_id	Integer Field	Foreign Key
4	Course_id	Integer Field	Foreign Key
5	Block_id	Integer Field	Foreign Key
6	Category_id	Integer Field	Foreign Key
7	Room_id	Integer Field	Foreign Key
8	Room_type_id	Integer Field	Foreign Key
9	Student_name	Char Field(200)	
10	Address	Char Field(200)	
11	State	Char Field(200)	
12	Street	Char Field(200)	
13	District	Char Field(200)	
14	Country	Char Field(200)	
15	Pincode	Integer Field()	
16	Guardian_name	Char Field(200)	
17	Student_phone	Integer Field()	
18	Guardian_phone	Integer Field()	
19	Email	Char Field(200)	
20	Relegion	Char Field(200)	

21	Caste	Char Field(200)	
22	Concession	Char Field(200)	
23	DOB	Integer Field()	
24	Distance_from_home	Integer Field()	
25	Year_of_admision	Integer Field()	
26	Room_type	Char Field(200)	
27	Room_no	Char Field(200)	
28	Documents	File Field()	
29	Photo	File Field()	
0	Course_name	Char Field(200)	
31	Department_name	Char Field(200)	
32	Block_name	Char Field(200)	
33	Category_name	Char Field(200)	
34	Place	Char Field(200)	
35	Gender	Char Field(200)	
36	Admission_no	Integer Field()	
37	Post	Char Field(200)	

28. Table Name: Feedback

Sl.No	Field	Data type	Data type
1	Feedback _id	Integer Field	Primary Key
2	Department_id	Integer Field	Foreign Key
3	Student_id	Integer Field	Foreign Key
4	Date	Date field()	
5	Feedback_title	Char Field(100)	
6	Department_name	Char Field(100)	
7	Student_name	Char Field(100)	
8	Feedback	Char Field(500)	

SYSTEM TESTING

SYSTEM TESTING

Testing is the major quality measure employed during software development. After the coding phase, computer programs are available that can be executed for testing purposes. Testing not only has to uncover errors introduced during coding but also locates errors committed during the previous phases. Thus, the aim of testing is to uncover requirements, design or coding in the program. System testing is an expensive but critical process that can take as much as fifty percent of the budget for program development. Consequential, different levels of testing are employed. In fact a successful test is one that finds an error. The system performance criteria deals with turnaround time backup, file protection and human factor. A test for the user acceptance should be carried out. The package developed was taken through different levels of testing and required modifications were made. Testing is a vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The following points show how testing is essential.

- Existence of program defects of inadequacies is inferred
- Verifies whether the software behave as intended by its designer.
- Checks conformance with requirements specification/user needs.
- Assesses the operational reliability of the system.
- Test the performance of the system.
- Reflects the frequencies of actual user inputs.
- Find the fault which caused the output anomaly.
- Detect flaws and deficiencies in the requirements.
- Exercise the program using data like real data processed by the program.
- Test the system capabilities.

TYPES OF TESTING

System testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests.

- Unit Testing
- Integration Testing
- Validation Testing
- Input Testing
- Output Testing
- User Acceptance Testing

Unit Testing

Unit testing focuses on the verification effort on the smallest unit of software design the software component module. Using the component level design as a guide, important control paths are tested to uncover the error within in the boundary of the module. The relative complexity of tests and uncovered error is limited by the constrained scope established for unit testing. Each module was tested individually and the errors are corrected.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit test components and build a program structure that has been dictated by design. Each module after unit testing were integrated and tested and errors were fixed.

Validation Testing

Here the inputs are given by the user are validated. This is the password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing. While verification is quality control process, quality assurance process carried out before the software is ready for release is known as validation testing. Its goal is to validate and be confident about the software product or system, that fulfills the requirements given by the customer. The two major areas when it should take place are in the early stages of software development and towards the end, when the product is ready for release. In other words, it is acceptance testing which is a part of validation testing.

Input Testing

Here system is tested with all verifiable combination of inputs. User may type data in situations like entering password, numerical details etc. The system is tested with all the cases and it responded error messages.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output generator or displayed by the system under consideration is tested by asking the user about the format required by them. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in term with the user requirement. Hence output testing does not result any correction in the system.

User Acceptance Testing

User acceptance testing is done in presence of user. User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly in touch with the prospective system users at time of developing and making changer wherever is done in regard to the following points:

- Input screen design
- Output screen design
- Menu driven system

FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

- Future enhancement of hostel Management system focuses on hardware implementation
- such as bar code reading system for marking student attendance.
- This hostel management software is designed for people who want to manage various activities in the hostel and fulfill the need of the future generation.
- We have expected that it will be helpful to the Customers as well as administrative member and we will improve this project by providing a Punching card to permanent students for automatic check in and check out.
- Future enhancement of the system focusses on implementation of gateway payment system.
- It is better to suggest Admission number as Student id for a centralized software implementation.
- We are waiting for your best suggestion and encouragement which could make us improve the future programming much better than the one we have carried out

CONCLUSION

CONCLUSION

To conclude the description about the project, developed in Python Django framework is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

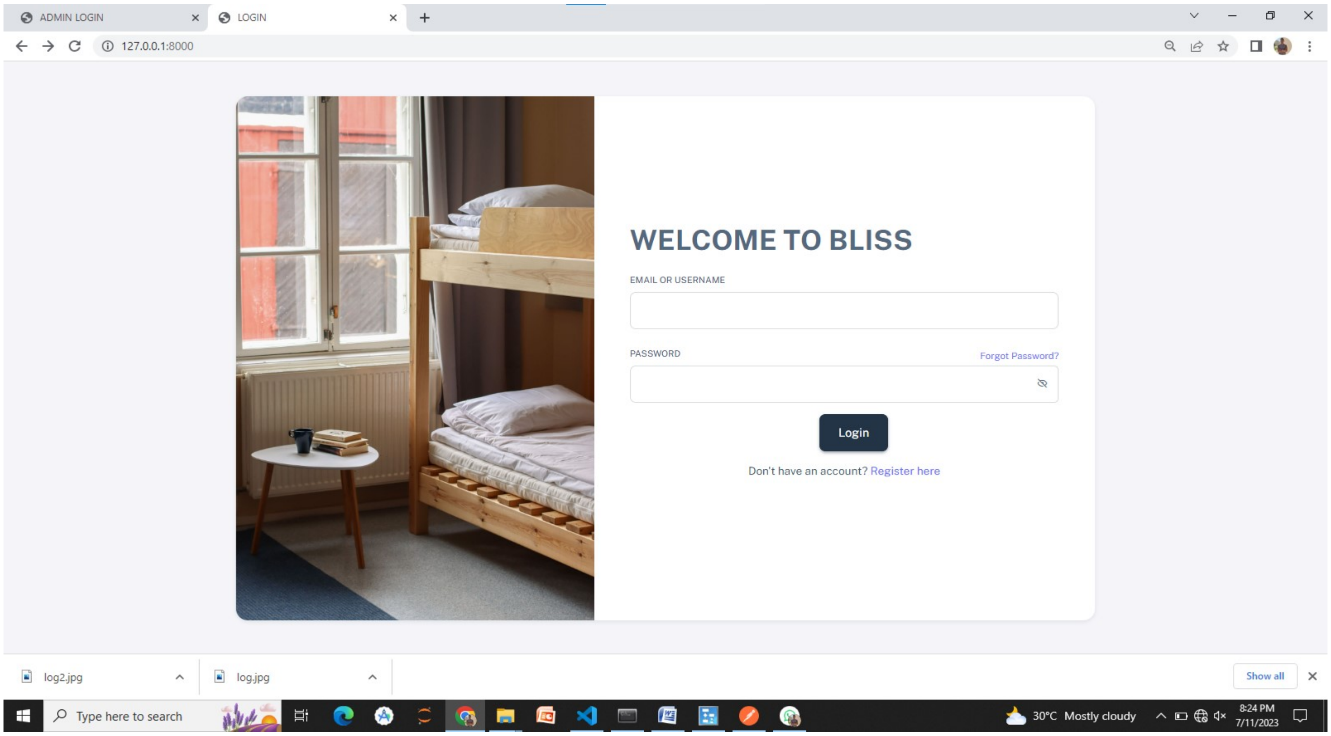
HOSTEL MANAGEMENT SYSTEM is very useful for hostel allotment and mess fee calculation. This hostel management software is designed for people who want to manage various activities in the hostel.

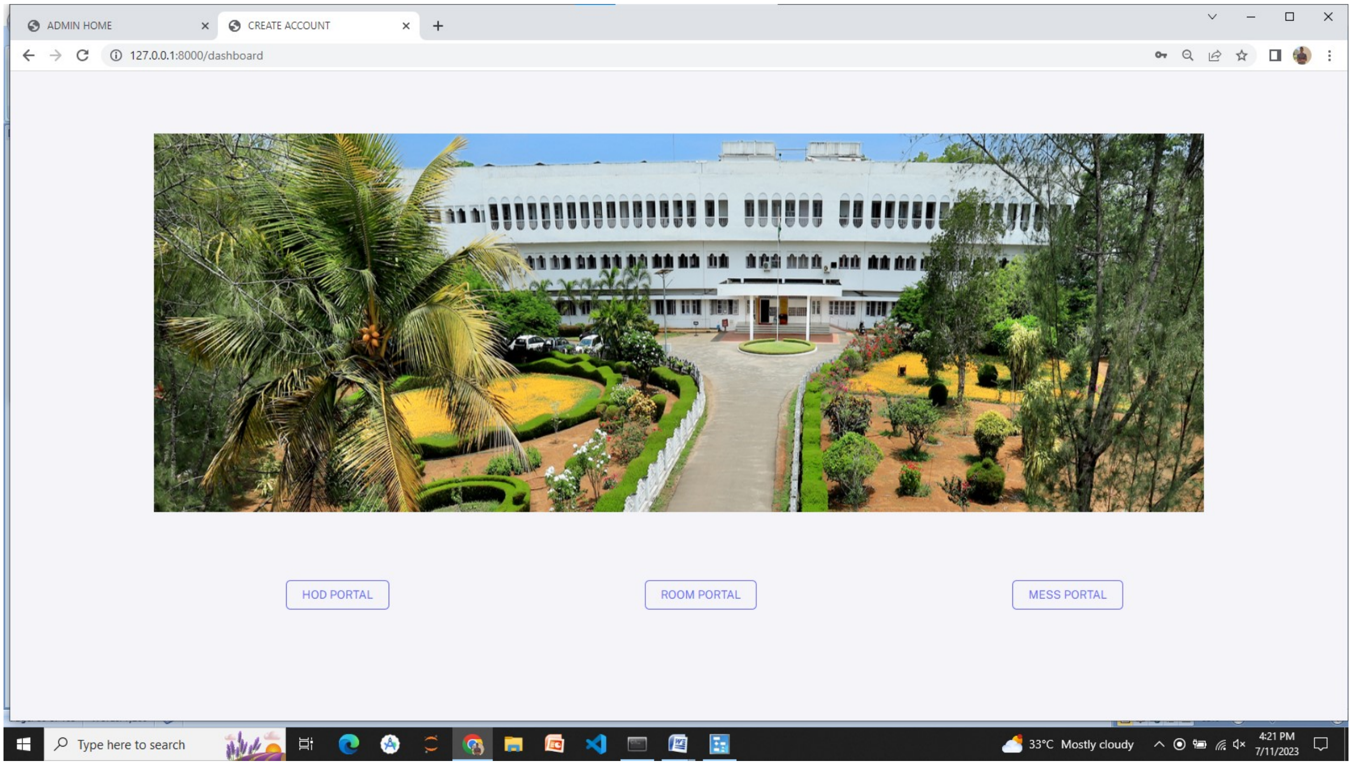
This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

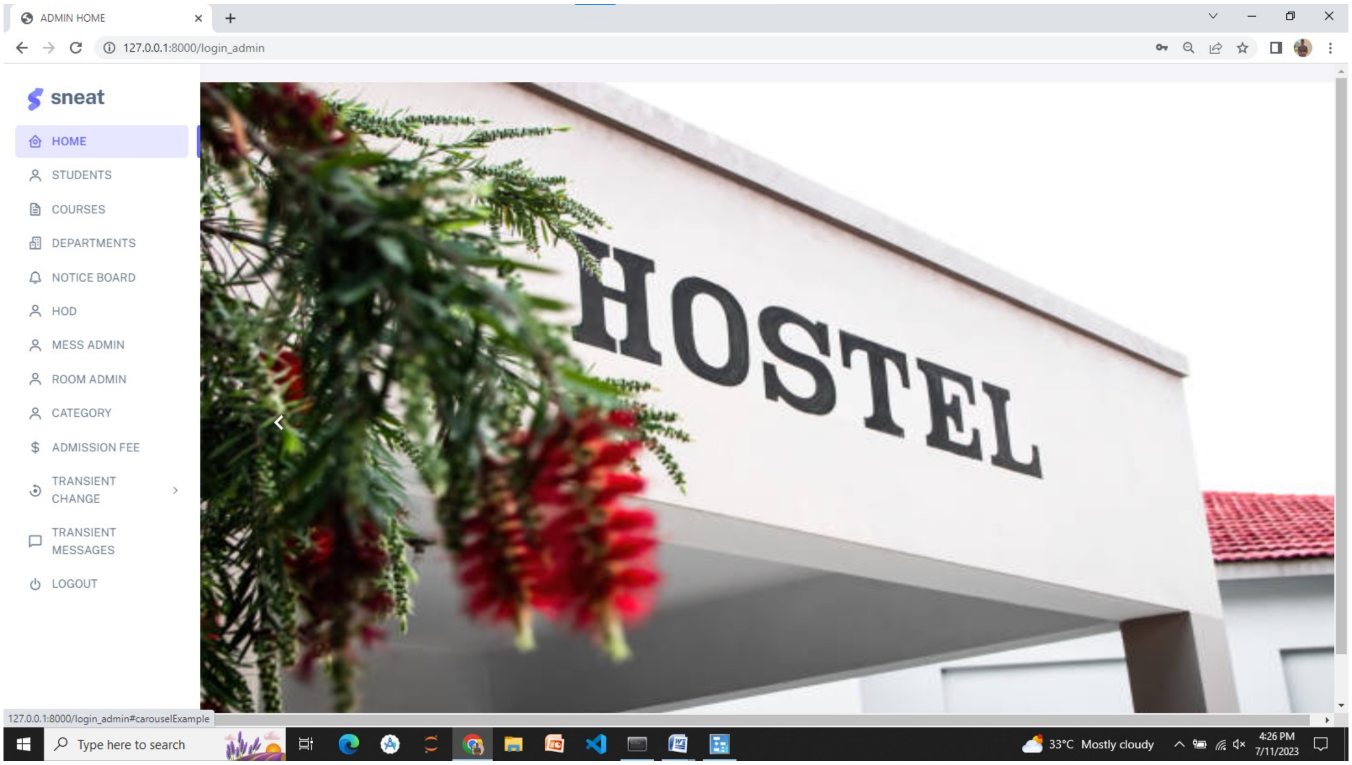
Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

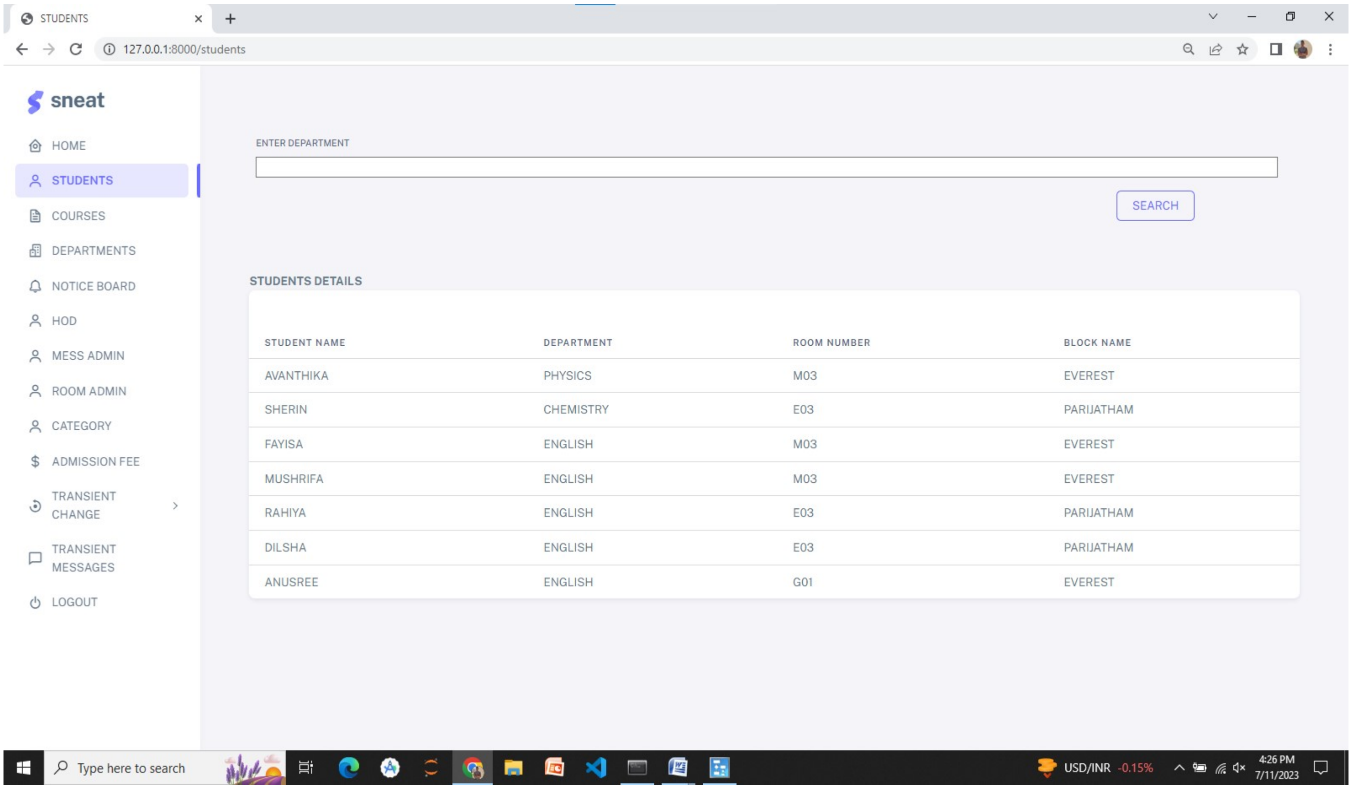
APPENDIX

SCREEN SHOTS



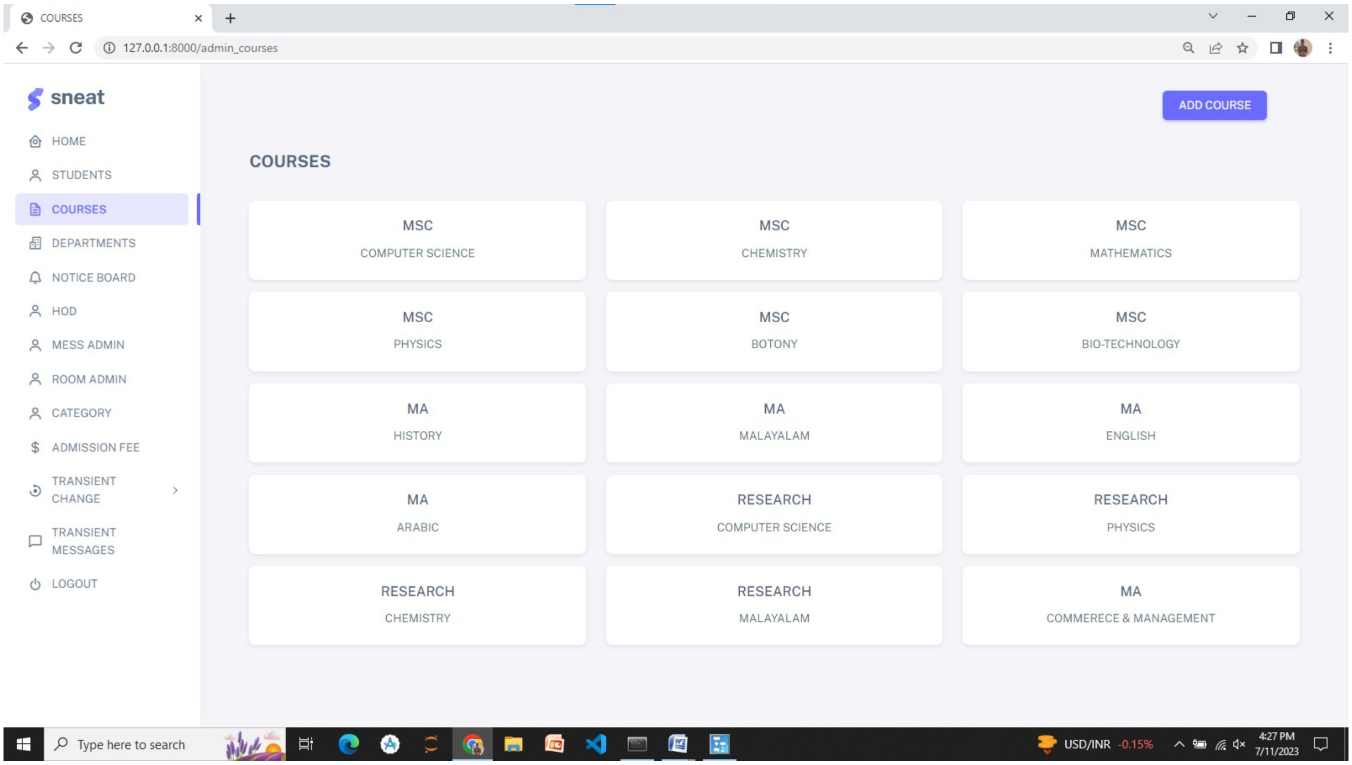






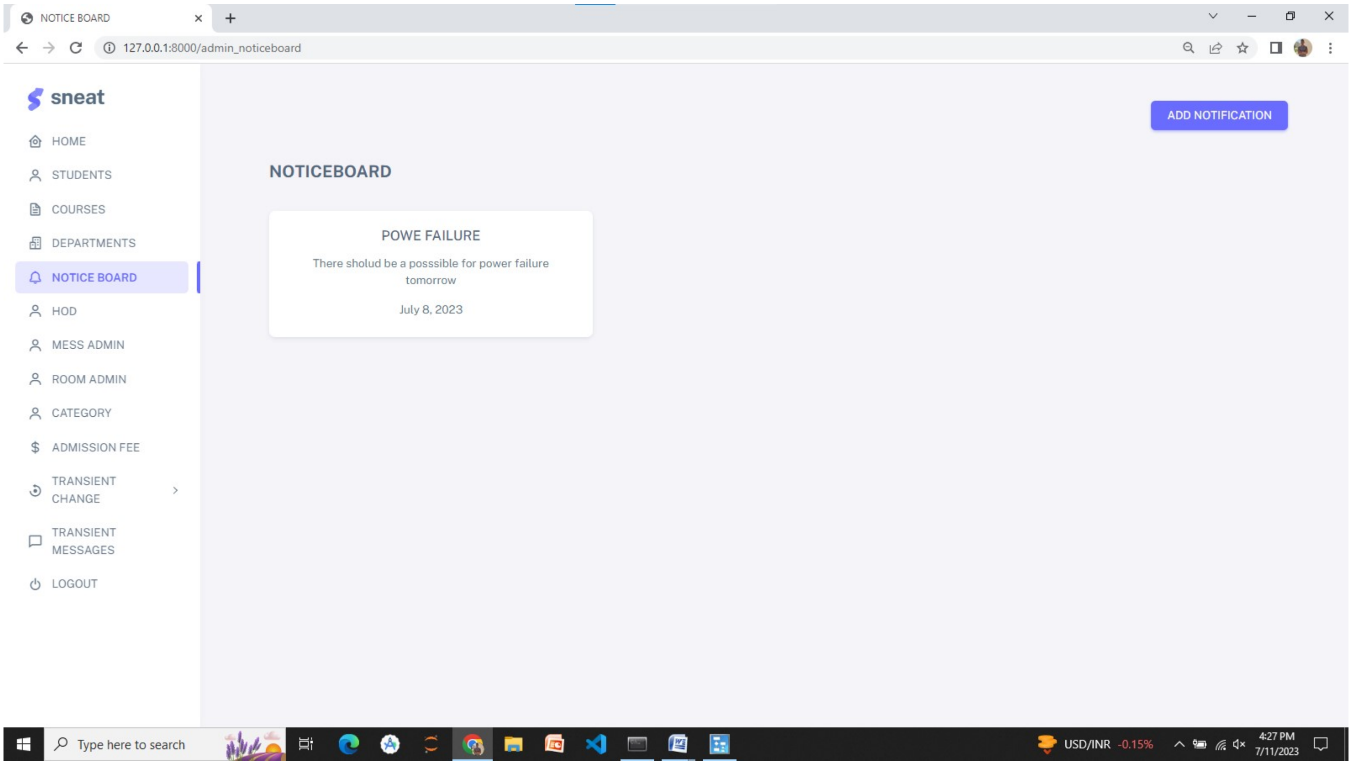
STUDENTS DETAILS

STUDENT NAME	DEPARTMENT	ROOM NUMBER	BLOCK NAME
AVANTHIKA	PHYSICS	M03	EVEREST
SHERIN	CHEMISTRY	E03	PARIJATHAM
FAYISA	ENGLISH	M03	EVEREST
MUSHRIFA	ENGLISH	M03	EVEREST
RAHIYA	ENGLISH	E03	PARIJATHAM
DILSHA	ENGLISH	E03	PARIJATHAM
ANUSREE	ENGLISH	G01	EVEREST



COURSES

MSC COMPUTER SCIENCE	MSC CHEMISTRY	MSC MATHEMATICS
MSC PHYSICS	MSC BOTONY	MSC BIO-TECHNOLOGY
MA HISTORY	MA MALAYALAM	MA ENGLISH
MA ARABIC	RESEARCH COMPUTER SCIENCE	RESEARCH PHYSICS
RESEARCH CHEMISTRY	RESEARCH MALAYALAM	MA COMMERECE & MANAGEMENT

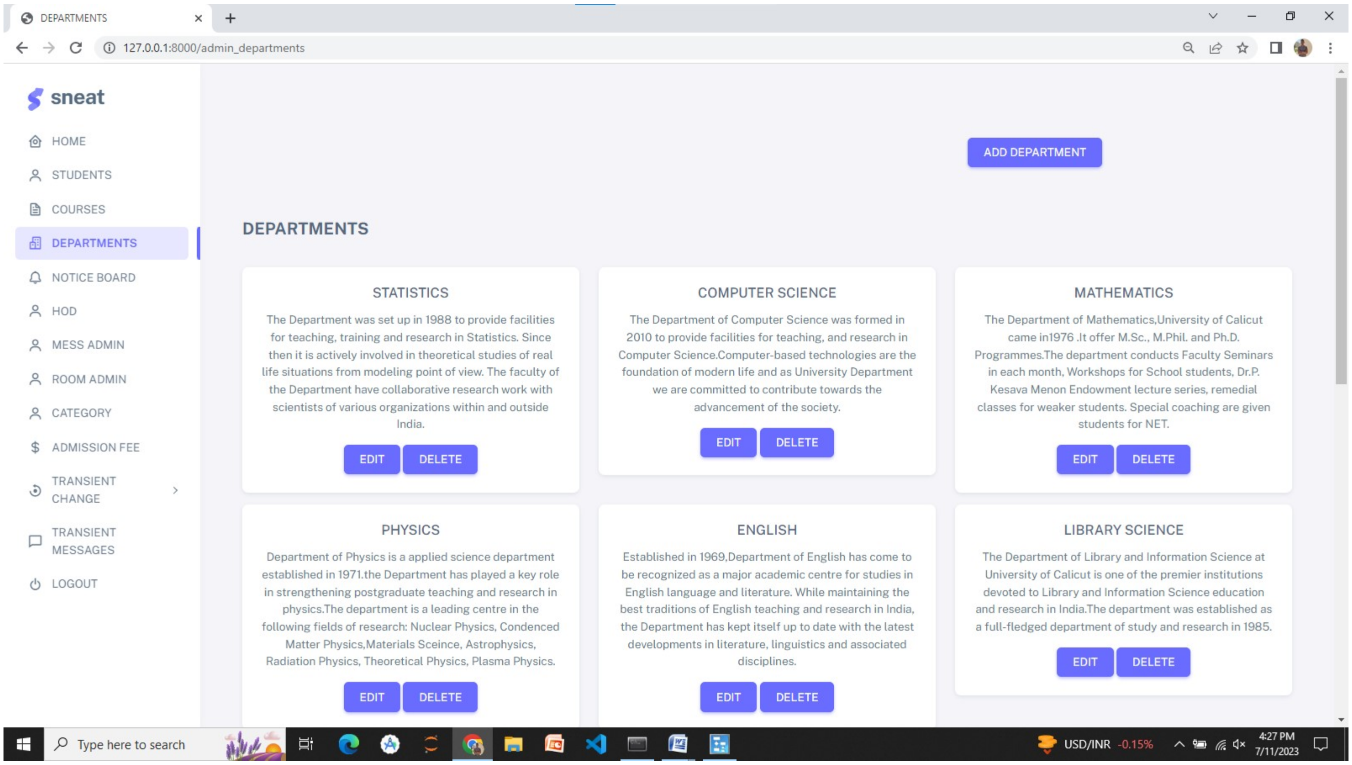


NOTICEBOARD

POWE FAILURE

There sholud be a posssible for power failure tomorrow

July 8, 2023



DEPARTMENTS

STATISTICS

The Department was set up in 1988 to provide facilities for teaching, training and research in Statistics. Since then it is actively involved in theoretical studies of real life situations from modeling point of view. The faculty of the Department have collaborative research work with scientists of various organizations within and outside India.

COMPUTER SCIENCE

The Department of Computer Science was formed in 2010 to provide facilities for teaching, and research in Computer Science.Computer-based technologies are the foundation of modern life and as University Department we are committed to contribute towards the advancement of the society.

MATHEMATICS

The Department of Mathematics,University of Calicut came in1976 .It offer M.Sc., M.Phil. and Ph.D. Programmes.The department conducts Faculty Seminars in each month, Workshops for School students, Dr.P. Kesava Menon Endowment lecture series, remedial classes for weaker students. Special coaching are given students for NET.

PHYSICS

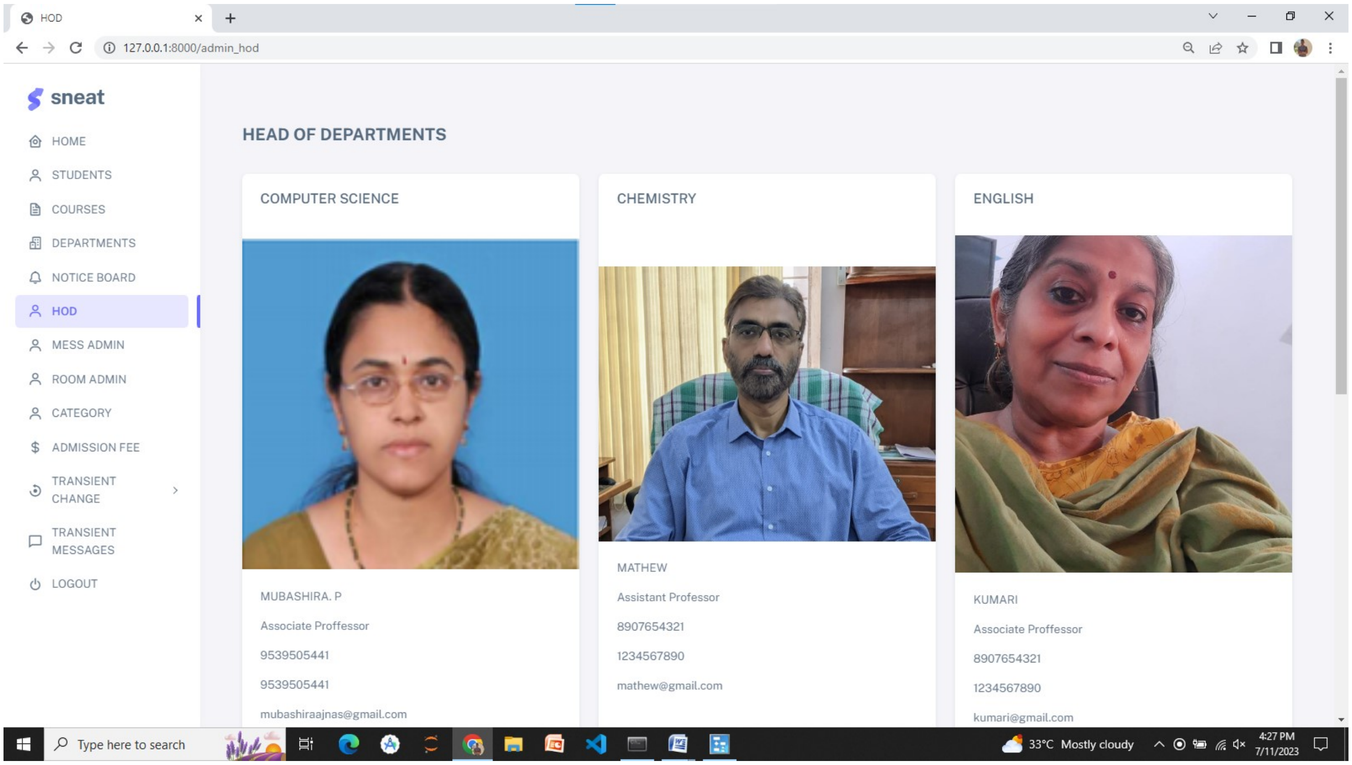
Department of Physics is a applied science department established in 1971.the Department has played a key role in strengthening postgraduate teaching and research in physics.The department is a leading centre in the following fields of research: Nuclear Physics, Condenced Matter Physics,Materials Sceince, Astrophysics, Radiation Physics, Theoretical Physics, Plasma Physics.

ENGLISH

Established in 1969,Department of English has come to be recognized as a major academic centre for studies in English language and literature. While maintaining the best traditions of English teaching and research in India, the Department has kept itself up to date with the latest developments in literature, linguistics and associated disciplines.

LIBRARY SCIENCE

The Department of Library and Information Science at University of Calicut is one of the premier institutions devoted to Library and Information Science education and research in India.The department was established as a full-fledged department of study and research in 1985.



HEAD OF DEPARTMENTS

COMPUTER SCIENCE

MUBASHIRA. P

Associate Proffessor

9539505441

9539505441

mubashiraajnas@gmail.com

CHEMISTRY

MATHEW

Assistant Professor

8907654321

1234567890

mathew@gmail.com

ENGLISH

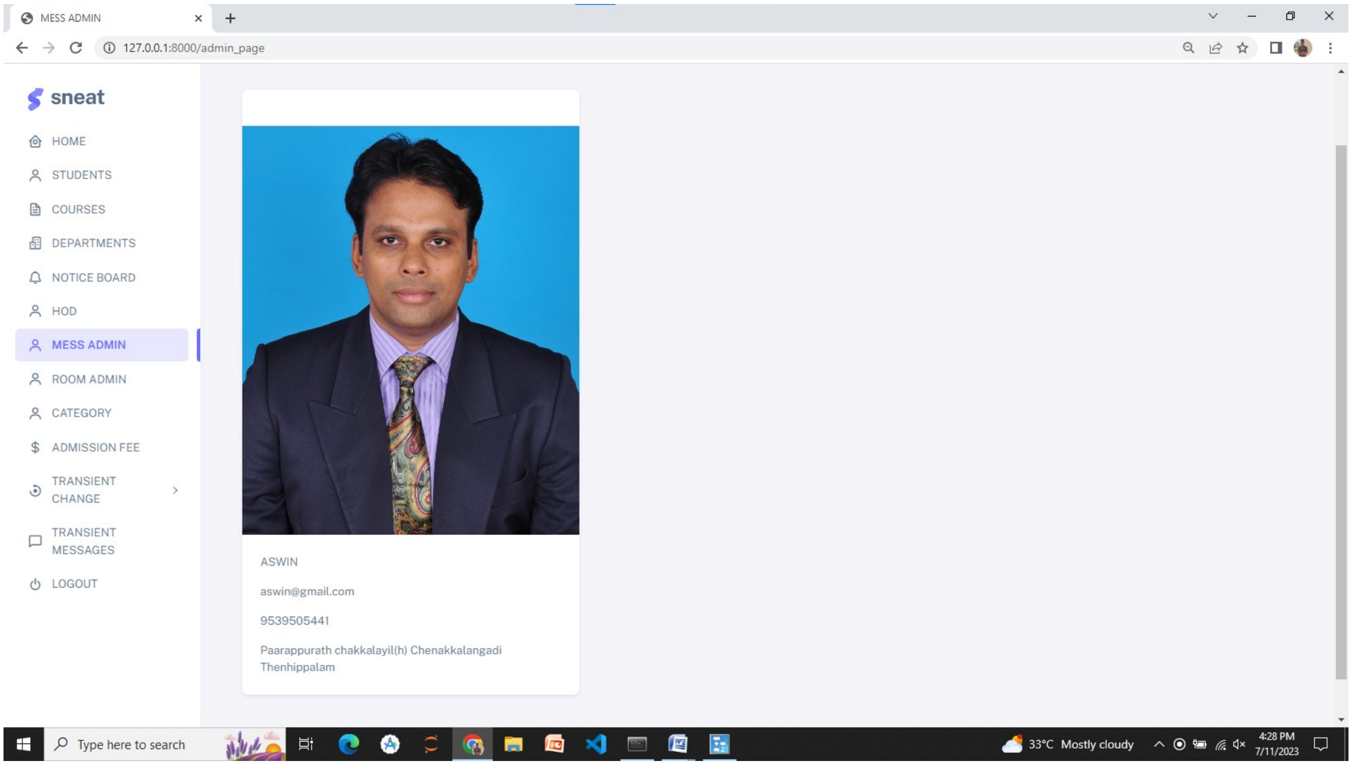
KUMARI

Associate Proffessor

8907654321

1234567890

kumari@gmail.com

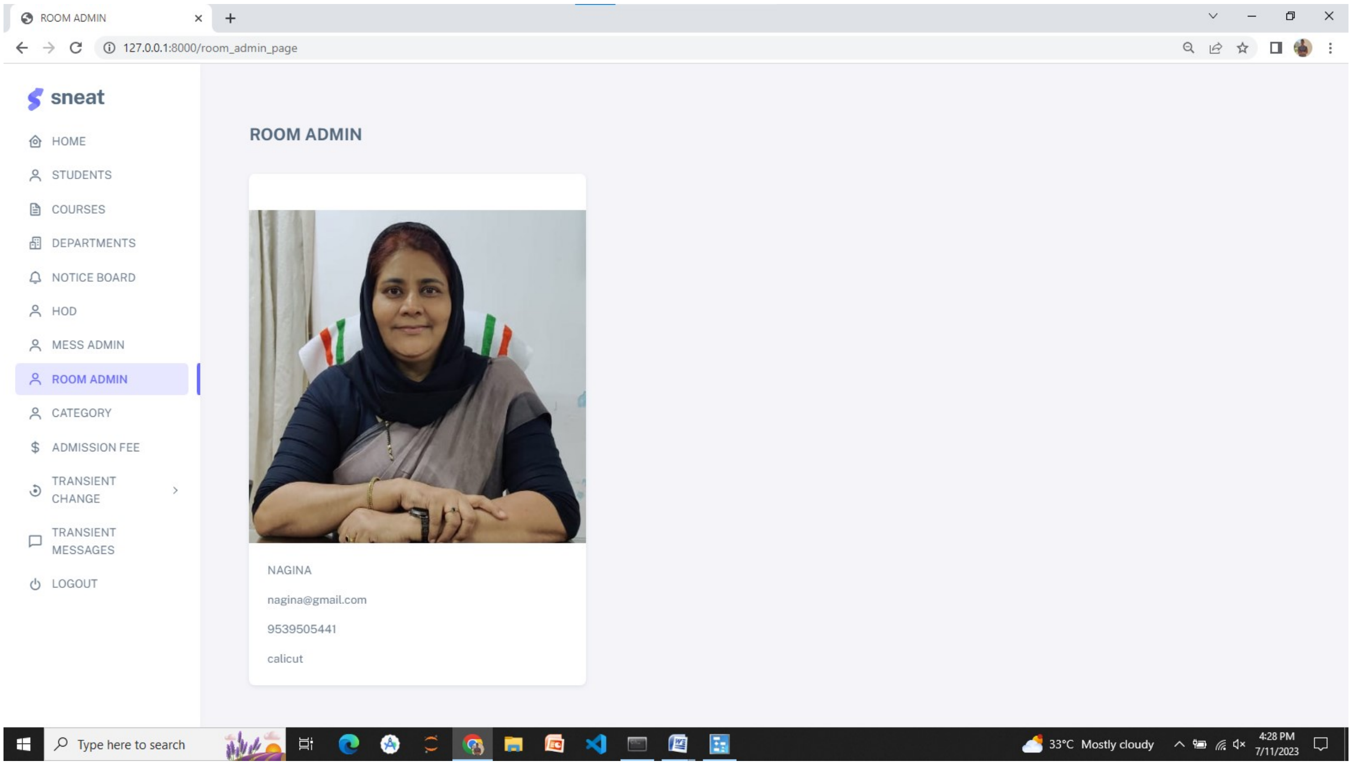


ASWIN

aswin@gmail.com

9539505441

Paarappurath chakkalayil(h) Chenakkalangadi Thenhippalam



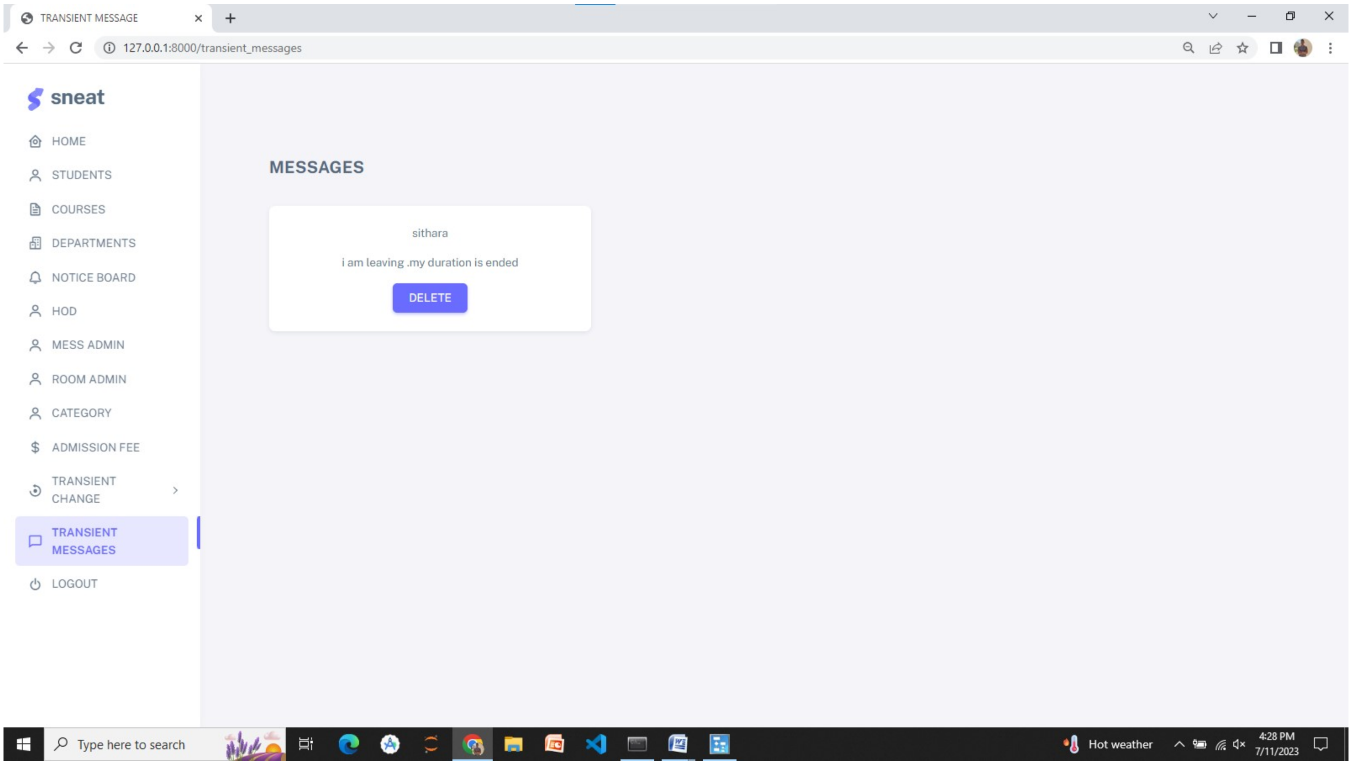
ROOM ADMIN

NAGINA

nagina@gmail.com

9539505441

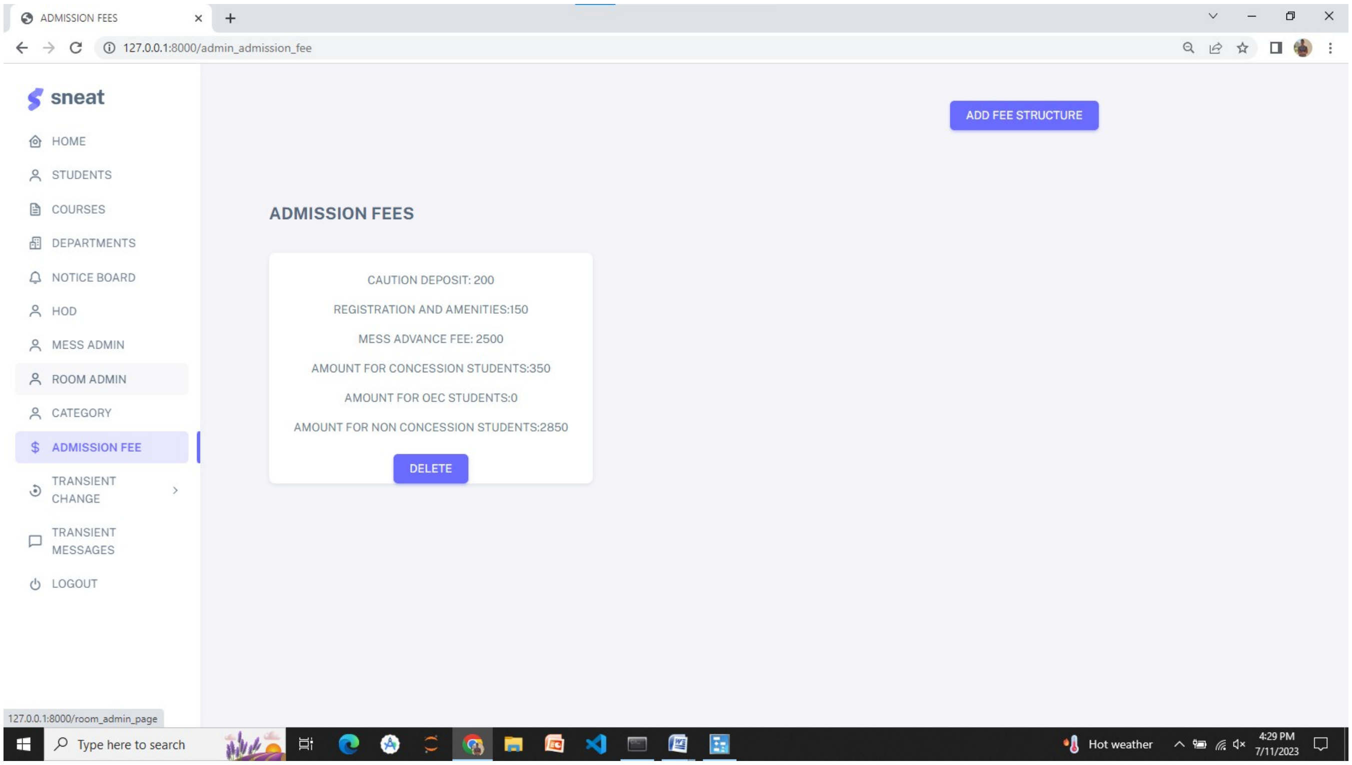
calicut



MESSAGES

sithara

i am leaving .my duration is ended



ADMISSION FEES

CAUTION DEPOSIT: 200

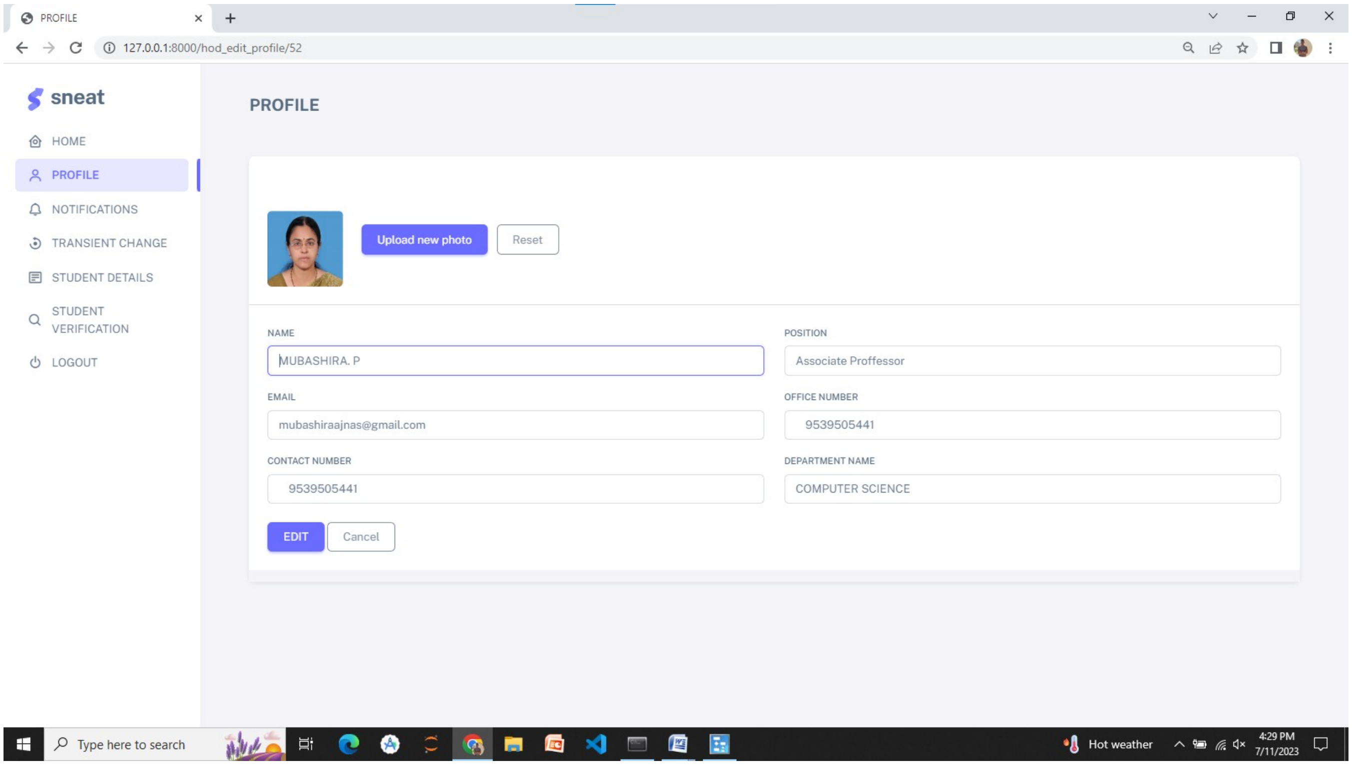
REGISTRATION AND AMENITIES:150

MESS ADVANCE FEE: 2500

AMOUNT FOR CONCESSION STUDENTS:350

AMOUNT FOR OEC STUDENTS:0

AMOUNT FOR NON CONCESSION STUDENTS:2850



PROFILE

NAME: MUBASHIRA. P

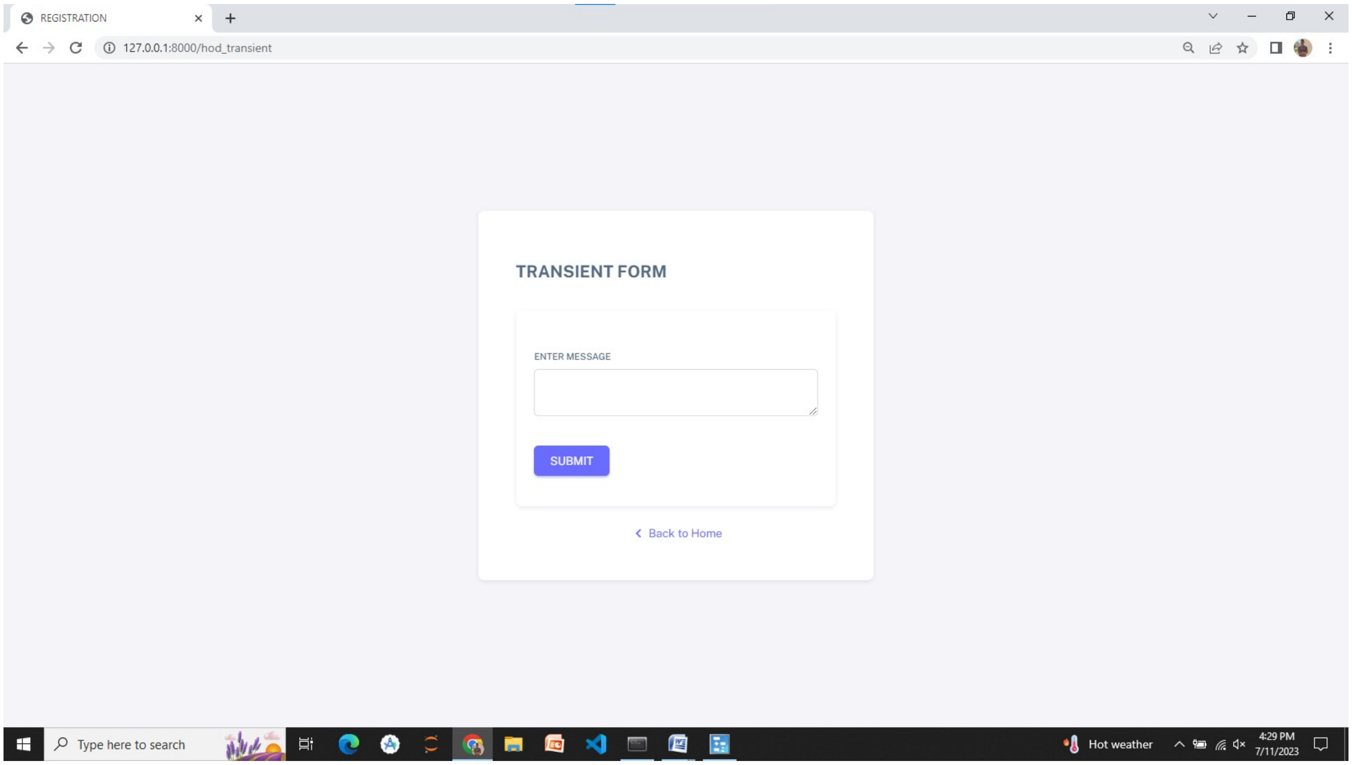
POSITION: Associate Proffessor

EMAIL: mubashiraajnas@gmail.com

OFFICE NUMBER: 9539505441

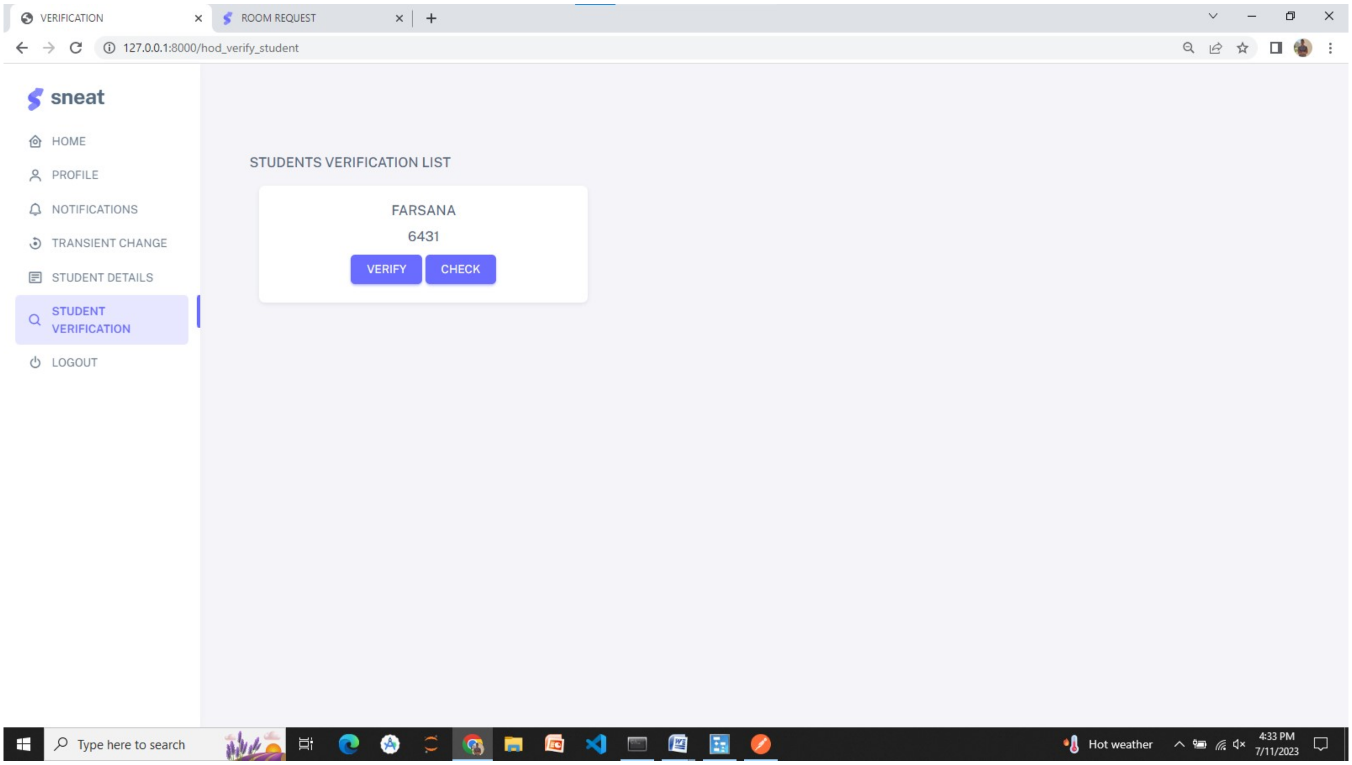
CONTACT NUMBER: 9539505441

DEPARTMENT NAME: COMPUTER SCIENCE



TRANSIENT FORM

ENTER MESSAGE



STUDENTS VERIFICATION LIST

FARSANA

6431

SAMPLE CODES

ADMIN

1.Courses.html

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#basicModal">

ADD COURSE

</button>



<!-- Modal -->

 <div class="modal fade" id="basicModal" tabindex="-1" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel1">Add Course Details</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form action="{% url 'admin_add_course' %}" method="post">

{% csrf_token %}

<div class="modal-body">

<div class="row">

<div class="mb-3">
```

```
<label for="exampleFormControlSelect1" class="form-label">Select Cource
Catogory</label>
 <input type="text" id="emailBasic" class="form-control" placeholder="Enter course"
 name="course" />
</div>
</div>
<div class="row g-2">
<div class="col mb-0">
<label for="emailBasic" class="form-label">Department</label>
<input type="text" id="emailBasic" class="form-control" placeholder="Enter
department"
name="department" />
</div>

</div>
  </div>
 <div class="modal-footer">
<button type="button" class="btn btn-outline-secondary" data-bs-dismiss="modal">
 Close
 </button>
<button type="submit" class="btn btn-primary">Save changes</button></div>
</form>
</div>
```

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

<div class="container-xxl flex-grow-1 container-p-y">
<h4 class="fw-bold py-3 mb-4"><span class="text-muted fw-light"></span>COURSES</h4>
<div class="row mb-5">
{% for i in data %}
<div class="col-md-6 col-lg-4">
<div class="card text-center mb-3">
<div class="card-body">
<h5 class="card-title">{{i.Course_name}}</h5>
<p class="card-text">{{i.Department_name}}</p>
</div>
</div>
</div>
{% endfor %}
</div>
</div>
</div></div>
</div></div>
```

1.1 Course view function

```
def admin_add_course(request):
    if request.method == 'POST':
        dpt = request.POST.get('department')
        print(dpt)
        course_name = request.POST.get('course')
        status = "0"

        dept = Department.objects.filter(Department_name=dpt).first()
        print(dept)

        if dept:
            courseDetail = models.Course(dpt=dept, Course_name=course_name,
Department_name=dept.Department_name, Status=status)
            courseDetail.save()
            print(courseDetail)
            print('course added')
            return render(request, 'admins/courses.html')

        print('Department not found')
        return render(request, 'admins/courses.html')
else:
        return render(request, 'admins/courses.html')
```

2.Department.html

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-
target="#basicModal">
ADD DEPARTMENT
</button>

<!-- Modal -->
<div class="modal fade" id="basicModal" tabindex="-1" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
 <h5 class="modal-title" id="exampleModalLabel1">Add Department Details</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<form action="{% url 'admin_add_department' %}" method="post">
{% csrf_token %}
<div class="modal-body">
<div class="row">
<div class="col mb-3">
<label for="nameBasic" class="form-label">Department name</label>
 <input type="text" id="nameBasic" class="form-control" placeholder="Enter Department
name"
name="department" />
 </div>
 </div>
<div class="row g-2">
<div>
<label for="exampleFormControlTextarea1" class="form-label">Department
Description</label>
 <textarea class="form-control" id="exampleFormControlTextarea1" rows="5"
 name="description"></textarea>
</div>

</div>
</div>
<div class="modal-footer">
 <button type="button" class="btn btn-outline-secondary" data-bs-dismiss="modal">
Close
</button>
  <button type="submit" class="btn btn-primary">Save changes</button>
 </div>
</form>
</div>

<!-- Text alignment -->

<div class="container-xxl flex-grow-1 container-p-y">
<h4 class="fw-bold py-3 mb-4">
 <span class="text-muted fw-light"></span>DEPARTMENTS
</h4>
 <div class="row mb-5">
{% for i in data %}
<div class="col-md-6 col-lg-4">
 <div class="card text-center mb-3">
<div class="card-body">
<h5 class="card-title">{{i.Department_name}}</h5>
<p class="card-text">
 {{i.Department_description}}
</p>
<a href="{% url 'admin_edit_department' i.id %}" class="btn btn-primary">EDIT</a>
<a href="{% url 'admin_delete_department' i.id %}" class="btn btn-primary">DELETE</a>
</div>
</div>
</div>
{% endfor %}

</div>
```

2.1 Department view function

```
def admin_add_department(request):

    if request.method == 'POST':



        department_name = request.POST.get('department')

        department_description = request.POST.get('description')

        status="0"
        deptDetail = models.Department(Department_name=department_name ,
Department_description=department_description, Status=status)

        deptDetail.save()

        print(deptDetail)

        print('department added')

        return render(request,'admins/departments.html')

    else:

        return render(request,'admins/departments.html')
```

3.Noticeboard.html

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#basicModal">

ADD NOTIFICATION

 </button>

<div class="modal fade" id="basicModal" tabindex="-1" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

 <div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel1">Add Notification</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form action="{% url 'admin_add_notice' %}" method="post">

{% csrf_token %}

<div class="modal-body">

<div class="row">

<div class="col mb-3">
```

```
<label>Enter Title Of Notification</label>
 <textarea id="basic-default-message" class="form-control"
placeholder="Enter the notification title" name="title">
</textarea>
</div>
</div>
<div class="row">
<div class="col mb-3">
<label>Enter Content Of Notification</label>
<textarea id="basic-default-message" class="form-control"
placeholder="Enter the notification" name="content">
</textarea>
</div>
 <div class="modal-footer">
<button type="button" class="btn btn-outline-secondary" data-bs-dismiss="modal">
Close
 </button>
<button type="submit" class="btn btn-primary">POST</button>
</div>
 </form>
</div></div>
<br>
        <!-- <h5 class="pb-1 mb-4">Text alignment</h5> -->
 <div class="container-xxl flex-grow-1 container-p-y">
 <h4 class="fw-bold py-3 mb-4">
```

```
<span class="text-muted fw-light"></span>NOTICEBOARD</h4>
<div class="row mb-5">
 {% for i in data %}
<div class="col-md-6 col-lg-4">
 <div class="card text-center mb-3">
  <div class="card-body">
<h5 class="card-title">{{i.Title}}</h5>
 <p class="card-text">
{{i.Content}}
</p>
<p class="card-text">
 {{i.Date}}
 </p>
 </div>
</div>
</div>
{% endfor %}
</div>
```

3.1 Noticeboard view function

```
def admin_add_notice(request):
    if request.method == 'POST':
        user = request.POST.get('user')
        title = request.POST.get('title')
        content = request.POST.get('content')
        # Get the current date and time

        date = datetime.now().date()

        status = "0"


        noticeDetail = models.Noticeboard(user=user, Title=title, Content=content, Date=date,
Status=status)

        noticeDetail.save()

        print(noticeDetail)

        print('notice added')

        return render(request, 'admins/noticeboard.html')

    else:

        return render(request, 'admins/noticeboard.html')
```

HOD

4.Profile.html

```
<div class="card mb-4">


<h5 class="card-header"></h5>

 <!-- Account -->

<form action="{% url 'hodforumupdate' i.id %}" method="post" enctype="multipart/form-data">

 {% csrf_token %}

<div class="card-body">

 <div class="d-flex align-items-start align-items-sm-center gap-4">

<img
```

```
 alt="user-avatar"
class="d-block rounded"
 height="100"
width="100"
id="uploadedAvatar"
name="photo"
src="/media/{{i.Photo}}"
/>
 <div class="button-wrapper">
 <label for="upload" class="btn btn-primary me-2 mb-4" tabindex="0">
<span class="d-none d-sm-block">Upload new photo</span>
<i class="bx bx-upload d-block d-sm-none"></i>
<input
type="file"
 id="upload"
 class="account-file-input"
hidden
accept="image/png, image/jpeg"
  name="photo"
/>
 </label>
<button type="button" class="btn btn-outline-secondary account-image-reset mb-4">
<i class="bx bx-reset d-block d-sm-none"></i>
<span class="d-none d-sm-block">Reset</span>
</button>
```

```
<!-- <p class="text-muted mb-0">Allowed JPG, GIF or PNG. Max size of 800K</p> -->
</div>
</div>
</div>
<hr class="my-0" />
<div class="card-body">
<form id="formAccountSettings" method="POST" onsubmit="return false">
<div class="row">
<div class="mb-3 col-md-6">
<label for="firstName" class="form-label">NAME</label>
<input
 class="form-control"
type="text"
id="firstName"
value="{{i.Name}}"
autofocus
name="name"
/>
</div>
 <div class="mb-3 col-md-6">
<label for="lastName" class="form-label">POSITION</label>
<input class="form-control" type="text"  id="lastName"  name="position" value="{{i.Position}}"/>
</div>
<div class="mb-3 col-md-6">
```

```
<label for="email" class="form-label">EMAIL</label>
 <input
 type="text"
id="email"
value="{{i.Email}}"
placeholder=""
name="email"
 />
</div>
<div class="mb-3 col-md-6">
<label class="form-label" for="phoneNumber">OFFICE NUMBER</label>
<div class="input-group input-group-merge">
<span class="input-group-text"></span>
<input
type="text"
id="phoneNumber"
  class="form-control"
placeholder=""
name="office"
 value="{{i.Office_no}}"
   </div>
  </div>
<div class="mb-3 col-md-6">
<label class="form-label" for="phoneNumber">CONTACT NUMBER</label>
<div class="input-group input-group-merge">
```

```
<span class="input-group-text"></span>

<input

type="text"

id="phoneNumber"

class="form-control"

placeholder=""

name="phone"

value="{{i.Contact_no}}

/>

</div>

</div>


<div class="mb-3 col-md-6">

<label for="state" class="form-label">DEPARTMENT NAME</label>

<input class="form-control"

type="text" id="state"

placeholder=""

name="department"

value="{{i.Department_name}}"/>

 </div>


 <div class="mt-2">

  <button class="btn btn-primary" >EDIT</button>

  <button type="reset" class="btn btn-outline-secondary">Cancel</button>

</div>
```

```
</form>

</div>
```

4.1 Profile view function

```
def hod_register(request):

    if request.method == 'POST' and request.FILES:


        name = request.POST.get('name')

        print('name')

        position = request.POST.get('position')

        email = request.POST.get('email')

        office = request.POST.get('office')

        contact = request.POST.get('phone')

        department = request.POST.get('department')

        username = request.POST.get('username')

        password1 = request.POST.get('password1')

        password2 = request.POST.get('password2')

        photo = request.FILES['photo']

        proof = request.FILES['identity']

        role='hod'

        status="0"

        dept = Department.objects.filter(Department_name=department).first()
        if password1==password2:
            if User.objects.filter(username=username).exists():


                return redirect('hod_portal')
            elif User.objects.filter(email=email).exists():


                return redirect('hod_portal')
            else:
                user=User.objects.create_user(username=username,password=password1)
                user.save()
                print(user)


                userDetail = models.Hod_registration(user=user,
dpt=dept,Department_name=dept.Department_name, Name=name, Position=position , Email=email,
Office_no=office,
                Contact_no=contact, Photo=photo, Proof=proof, Role=role, Status=status)
                userDetail.save()
                print(userDetail)


                print('user created')
        else:
              return redirect('hod_portal')
return redirect('login_page')
    else:
        return render(request, 'admins/hodlogin.html')
```

5.Verification.html

```
<h5 class="pb-1 mb-4">STUDENTS VERIFICATION LIST</h5>
<div class="row mb-5">
{% for i in data %}
{% if i.Status == "2" %}
<div class="col-md-6 col-lg-4">
 <div class="card text-center mb-3">
<div class="card-body">
 <h5 class="card-title">{{i.Student_name}}</h5>
 <h5 class="card-title">{{i.Admission_no}}</h5>
<!-- <p class="card-text">
   The last date to submit online application isextended up to next month
  </p> -->
 <button class="btn btn-primary" data-bs-target="#modalToggle2" data-bs-toggle="modal"
 data-bs-dismiss="modal">
<a class="text-white" href="{% url 'hod_send_verification' i.id %}">VERIFY</a>
</button>


 <a href="{% url 'student_details' %}" class="btn btn-primary">CHECK</a>


 </div>
 </div>
 </div>
 {% endif %}
 {% endfor %}
```

5.1 Verification view function

```
def hod_verify_student(request):
    if request.user:
        user=request.user
        data = Hod_registration.objects.filter(user=user).values()
        for i in data:
            dept = i["Department_name"]
            print(dept)
        dta = Student.objects.filter(Department_name=dept).values()
        print(dta)
        return render(request,'hod/verification.html',{'data':dta})


def hod_send_verification(request,id):
    user = Student.objects.get(id=id)
    user.Status = 3
    user.save()
    return redirect('hod_verify_student')
```

BIBLIOGRAPHY

BIBLIOGRAPHY

- Elias M Awad, "**System Analysis and Design**", Galgotia Publications, Second Edition
- Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa Publishing House, Edition 1991.
- Sliberschatz, Koroth, Sudarshan, "Database System Design Concepts".
- Ghezzi jazayers and mandriolli, "Fundamentals of Software Engineering", Second Edition
- http://www.wikipedia.com
- http://www.google.com
- http://www.w3schools.com
- http://flutter.org
- www.chatgpt..com