

Department Management System

Project Report Submitted in the Partial Fulfilment
of the requirements for the Degree of

Bachelors of Technology

in

Computer Science and Engineering

by

Zamin Ahmad Dar (18048112002)

Dar Ubaid Nazir (18048112009)

Under the Supervision of

Er. Khalid Hussain
(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF KASHMIR

NORTH CAMPUS

DELINA, BARAMULLA-193101

(SESSION 2022)

**University of Kashmir
North Campus
Delina, Baramulla 193101
2022**



Certificate

This is to certify that this project report entitled ***Department Management System*** by Zamin Ahmad Dar (18048112002) and Dar Ubaid Nazir (18048112009), submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of the University of Kashmir, North Campus, during the academic year **2022**, is a bonafide record of work carried out under our guidance and supervision.

(Guide)

(External Examiner)

(Head of Department)

STUDENTS' DECLARATION

We, Zamin Ahmad Dar (Roll No. 18048112002) and Dar Ubaid Nazir (Roll No. 18048112009), hereby declare that the project entitled ***“Department Management System”*** submitted in the partial fulfilment of the requirement for award of Bachelors of Technology in Computer Science and Engineering degree in the session 2022, is an authentic record of our own work carried out under the supervision of Er. Khalid Hussain, Department of Computer Science & Engineering, University of Kashmir, North Campus, Delina, Baramulla.

Zamin Ahmad Dar
(18048112002)

Dar Ubaid Nazir
(18048112009)

Dated: 12 December, 2022

ACKNOWLEDGEMENT

First and foremost, we would like to praise Allah the Almighty, the Most Gracious and the Most Merciful for his blessing and guidance which enabled us to complete the project *Department Management System*.

We wish to extend our greatest gratitude to the following people without whom the progress of the project could not have been made.

First of all, we are profoundly grateful to our teacher and guide Er. Khalid Hussain for his expert guidance and continuous encouragement throughout to see that this project reaches its target since its commencement to completion. We are also thankful to all other faculty members for their timely guidance that shall carry us a long way in the journey of life on which we are about to embark.

Also, we would like to put on record the support of our parents and classmates who always stood by us in situations of grim affliction and patted us when this work was going on.

Zamin Ahmad Dar
(18048112002)

Dar Ubaid Nazir
(18048112009)

ABSTRACT

This thesis on “*Department Management System*” is a complete analysis of development of web application which manages necessary information of departments in our university and makes this information more efficient, optimized and useful for all users by reducing time and efforts needed for management. The main goal of this web application is to manage all records related to students and teachers and then automate the system for document generation (attendance, lecture plan) for all the users like co-ordinator, teachers and students. This thesis contains the working details of all the modules that are present in this web application. It also describes all the web development tools and technologies that we used for developing this web application.

Table of Content

Chapter		Page No.
1.	INTRODUCTION.....	1-2
1.1.	MOTIVATION.....	1
1.2.	OBJECTIVES.....	1
1.3.	SCOPE.....	2
2.	LITERATURE REVIEW.....	3-4
2.1.	DEPARTMENT MANAGEMENT SYSTEM.....	3
2.2.	ONLINE COLLEGE MANAGEMENT SYSTEM.....	3
2.3.	COLLEGE DEPARTMENT MANAGEMENT SYSTEM.....	4
2.4.	STUDENT MANAGEMNT SYSTEM.....	4
3.	SYSTEM ANALYSIS AND DESIGN.....	5-14
3.1.	SYSTEM STUDY.....	5
3.2.	SYSTEM ANALYSIS.....	5
3.2.1.	USER REQUIREMENTS.....	5
3.2.2.	FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS.....	5
3.3.	SYSTEM DESIGN.....	6
3.3.1.	ITERATIVE INCREMENTAL MODEL.....	6
3.3.2.	SYSTEM ARCHITECTURE.....	8
3.3.3.	ER DIGRAM.....	8
3.3.4.	DFD.....	10
3.3.5.	ACTIVITY DIAGRAMS.....	12
4.	IMPLEMENTATION.....	15-25
4.1.	CLIENT SERVER ARCHITECTURE.....	15-16
4.1.1.	WORKING OF CLIENT SERVER ARCHITECTURE.....	16
4.1.2.	ADVANTAGES.....	16
4.1.3.	DISADVANTAGES.....	16
4.2.	HTTP.....	17
4.3.	HTTPS.....	19
4.4.	WEB DEVELOPMENT TECHNOLOGIES.....	19
4.5.	FRONT END TECHNOLOGIES.....	19
4.5.1.	HTML.....	20
4.5.2.	CSS.....	21
4.5.3.	JAVA SCRIPT.....	21
4.6.	BACKEND TECHGNOLOGIES.....	22
4.6.1.	PHP OBJECT ORIENTED.....	23
4.6.2.	AJAX.....	23
4.6.3.	SQL.....	24
4.7.	DATABASE.....	24
4.7.1.	MY SQL.....	24
4.8.	XAMPP.....	25
5.	SYSTEM TESTING.....	26-28
5.1.	COORDINATOR TEST CASE.....	26
5.2.	TEACHER TEST CASE.....	26
5.3.	STUDENT TEST CASE.....	26
5.4.	PDF GENERATION TEST CASE.....	26
6.	CODING SAMPLES.....	29-40
6.1.	COORDINATOR REGISTRATION.....	29
6.2.	COORDINATOR LOGIN.....	31
6.3.	JS CODE FOR SETTING LIMIT OF DAYS FOR ATTENDANCE.....	32
6.4.	ADDING A PROGRAM/BRANCH.....	33
6.5.	DELETE A SUBJECT.....	35
6.6.	GENERATE PDF OF ATTENDACE FOR ALL SUBJECTS.....	36
7.	SYSTEM INTERFACES.....	41-50
8.	CONCLUSION.....	51
8.1	FUTURE SCOPE.....	51
9.	REFERENCES.....	52

LIST OF FIGURES

Figures	Page No.
3.1 ITERATIVE INCREMENTAL MODEL	7
3.2 SYSTEM ARCHITECTURE	8
3.3 ER DIAGRAM	10
3.4 LEVEL 0 DFD.....	11
3.5 LEVEL 1 DFD.....	12
3.6 ACTIVITY DIAGRAM OF COORDINATOR MODULE	13
3.7 ACTUIVITY DIAGRAM OF TEACHER MODULE.....	14
3.8 ACTIVITY DIAGRAM OF STUDENT MODULE.....	14
4.1 CLIENT SERVER ARCHITECTURE	17
7.1 COORDINATOR LOGIN PAGE	41
7.2DASHBOARD PAGE WITH MENUS.....	41
7.3 ACTIVE AND CLOSED SEMESTERS.....	42
7.4 PDF GENERATION FOR A PARTICULAR SUBJECT.....	42
7.5 ENROLLED STUDENTS IN A BATCH	43
7.6 ATTENDANCE UPDATE REQUEST FORM	43
7.7 SYLLABUS DETAILS	44
7.8 LIST OF SUBJECTS.....	44
7.9 PDF OF ATTENDANCE DETAILS	45
7.10 ADD GROUPS AND SET WORKING TIME.....	45
7.11 SUBJECTS ASSIGNED TO A SEMESTER.....	46
7.12 SUBJECTS ASSIGNED TO A TEACHER.....	46
7.13 TEACHERS LIST	47
7.14 TEACHER LOGIN PAGE	48
7.15 TEACHER DASH BOARD PAGE.....	48
7.16 LIST OF SUBJECTS ASSIGNED TO A TEACHER	49
7.17 ATTENDANCE PAGE FOR THEORY SUBJECT	49
7.18 ATTENDACE PAGE FOR LAB SUBJECT	50
7.19 LECTURE DETAIL PAGE	50

LIST OF TABLES

Table	Page No.
TABLE 1 GENERAL TESTING CASES	27
TABLE 2 PDF GENERATION TESTING CASES	28

LIST OF ACRONYMS

URL	UNIFORM RESOURCE LOCATOR
DNS	DOMIAN NAME SYSTEM
IP	INTERNET PROTOCOL
HTTP	HYPERTEXT TRANSFER PTOTOCOL
HTTPS	HYPERTEXT TRANSFER PTOTOCOL SECURE
DOM	DOCUMENT OBJECT MODEL
CSS	CASCADING STYLE SHEETS
JS	JAVA SCRIPT
JIT	JUST IN TIME
TLS	TRANSPORT LAYER SECURITY
SSL	SECURE SOCKETS LAYER
SQL	STRUCTURED QUERY LANGUAGE
RDBMS	RELATIONAL DATABASE MANAGEMENT SYSTEM
SMS	SHORT MESSAGE SERVICE
PDF	PORTABLE DOCUMENT FORMAT
DFD	DATA FLOW DIAGRAM
SAS	SYNTATICAL AWESOME SHEETS
PDO	PHP DATA OBJECTS
AJAX	ASYNCHRONOUS JAVA SCRIPT AND XML

CHAPTER 1

INTRODUCTION

Digital technologies have advanced more rapidly than any innovation in our history. Technology handles any work much easily and smartly, making day-to-day dealings very convenient. In order to make the work of different fields in our department easier, more efficient, optimized and useful, there is a need to automate the work of various fields of our department so that majority of work which is currently based on pen and paper will be done digitally. So, taking into consideration the above idea we are proposing a management system which will be a common platform for all the members (students, faculty and coordinator/HOD) associated with our department to manage data in a very efficient way.

1.1 Motivation

Currently majority of official work in our department related to students, faculty members and co-ordinator is based on pen and paper and are entered manually. This manual based system is time consuming, inefficient and prone to errors. Such a system consumes large amount of physical space and is a tedious task. Additionally, to retrieve information from this data is also time consuming and is very prone to errors. So, taking into consideration the above limitations, we are proposing a management system which will be a common platform for all the members (students, faculty and coordinator/HOD) associated with our department to manage data in a very efficient way and remotely.

1.2 Objectives

Department Management System is a web-based application which allows admin (Co-ordinator) to access the information about his department including semesters, lab groups, subjects, students in each semester, teachers assigned to each subject, syllabus assigned to subjects, attendance of students, time conflict management for attendance, lecture plans etc. It

also provides support for faculty to update attendance after taking permission in case of mistake. Admin manages all the accounts of the student and teachers. Here we get the latest information about the students and teachers. Every teacher and student will have their own dashboards which they can access only after the proper authentication. It also automates document generation (attendance, lecture plan) for all the users like co-ordinator, teachers and students.

1.3 Scope

The scope of this project will focus on the co-ordinator, faculty members and students who will use the system to enter and retrieve information. This project will be implemented and useful for all the departments that have related needs as of our department. This system is managed mainly by the co-ordinator. This system manages complete details in a single application and in a single database. The users will use this system to handle all the functionalities easily. The intentions of the system are to reduce manual work and decrease the time taken to record data, keep it securely and retrieve the stored data. It is an open source application so that others can edit and transform this system application according to their needs can be a future enhancement in project.

CHAPTER 2

LITERATURE REVIEW

The details of some of research papers related to our work has been given below:

2.1 Department Management System

DMS is defined as an application build on web that is useful in providing information at all levels of a department. For the users of this system the administrator creates login IDs and respective passwords from which student/staff can easily access the system. From department library to department yearly magazine, from staff rating to student's feedback, every notice and upcoming events is showcased in this system. Placement and rooms/labs record are also stored in the system [1].

2.2 Online College Management System

Online College Management System (OCMS) provides a simple interface for maintenance of student information. Student information system deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details, placement details and other resource related details too. It will also have faculty details, batch execution details, students' details in all aspects, the various academic notifications to the staff and students updated by the college administration. It also facilitate us explore all the activities happening in the college, Different reports and Queries can be generated based on vast options related to students, batch, course, faculty, exams, semesters, certification and even for the entire college The placement officer is responsible for updating the placement related information like eligible criteria for a particular company, arriving date for the company which is coming for recruitment, the list of students who are eligible for attending the recruitment process. E-Library (also referred to as digital library) is a special library with a focused collection of digital objects [2].

2.3 College Department Management System

The system provides guidance to the admin to keep track of each student. The admin has the access to the database of system. In an educational institute management is crucial thing. So, in order to reduce the efforts of staff we are introducing our system. The system comes on with much functionality like voting event details, feedback, news line etc. It provides a additional feature newlines that helps the student to get department newlines and reports (achievements, toppers). It also provides the voting feature so that manual work is reduced. This system is paperless system. System provides functionality for student to application where in admin can manage, student can access uploaded notes, course details. Student will get the event details through SMS. Overall manpower and reduces the time required [3].

2.4 Student Management System

Student Management System manages several student details like USN, student attendance, internal assessment marks, parent name, phone number, email-id, date-of-birth, class, sex etc. The goal of evolving this application is to induce the report regarding attendance at the completion of the conclave or at the middle of the conclave. Also, it is possible to get the average of internal assessments and it is easy to get the report at the end. Faculty will update the student status by putting present or absent. Finally, student can only view his details, he can take the report. Student Management System has four modules. Initially admin will login, login module. Later he is going to upload the details of student, called student data module. At the end of the session report will be generated, called report module which is generated in the pdf format. If particular student is absent his status will be sent to the parent by a

SMS, called SMS module, and email will be sent to the respective parent and it is known as email module [4].

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

The chapter describes the system study, analysis (user requirement & system and hardware requirement), Entity Relationship Diagram, DFD and activity diagrams.

3.1 System Study

The study was carried out at admin and faculty level and the main purpose of the study was to find out how the process of recording students' data and data related to lectures is carried out. The system that is currently being used is entirely maintained manually. But we are creating an online system, that is more efficient than the existing manual system.

3.2 System Analysis

During the system study period, we can categorize into two parts our *Department Management System*, as a user requirement and system and hardware requirement.

3.2.1 User requirements

Referring to the existing system, we have investigated and found out how the current system operates, not only that but also tried out which problems are faced and how best they can be settled. The users described some of the basic requirements of the system that includes attendance management, lecture plan management, lab group management, syllabus management, view all types of reports and generate PDF documents for each report.

3.2.2 Functional and Non-Functional Requirements

Functional requirements

- Co-ordinators should be able to create an account for his department.
- Co-ordinator can create users' accounts for faculty members and can add students into a particular batch either by uploading an excel file containing emails of all the students or individually.

- Co-ordinator can add batches, branches, subjects and can assign subjects to teachers.
- Teacher can upload attendance and lecture plan of assigned subjects.
- Students can view their attendance details and can also get details of lecture plans.
- Co-ordinator can get a detailed view of the records provided by teachers.

Non-functional requirements

- Responsive and user-friendly UI
- Speed
- Less weight
- Reliability

3.3 System Design

Design is the first step in the development stage. Software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this part, because in this activity, decisions finally affecting the success of the software implementation and its ease of maintenance. Design is the only way to correctly translate the customer requirements into finished software or a system. Design is the place where quality is bringing up in development.

3.3.1 Iterative Incremental Model

The project is based on Object Oriented approach using Iterative Incremental Model. Iterative and incremental software development is a method of software development that is modelled around a gradual increase in feature additions and a cyclical release and upgrade pattern.

Iterative and incremental software development begins with planning and continues through iterative development cycles involving continuous user feedback and the incremental addition of features concluding with the deployment of completed software at the end of each cycle.

Iterative and incremental development is a discipline for developing systems based on producing deliverables. In incremental development, different parts of the system are developed at various times or rates and are integrated based on their completion. In iterative development, the plan is to revisit parts of the system in order to revise and improve them. User feedback is consulted to modify the targets for successive deliverables.

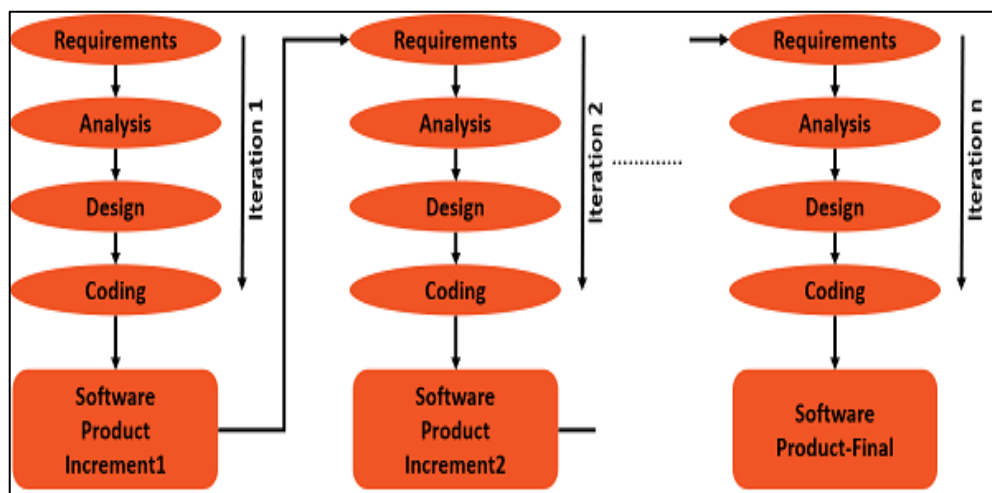


Fig.3.1 Iterative Incremental Model

3.3.2 System Architecture

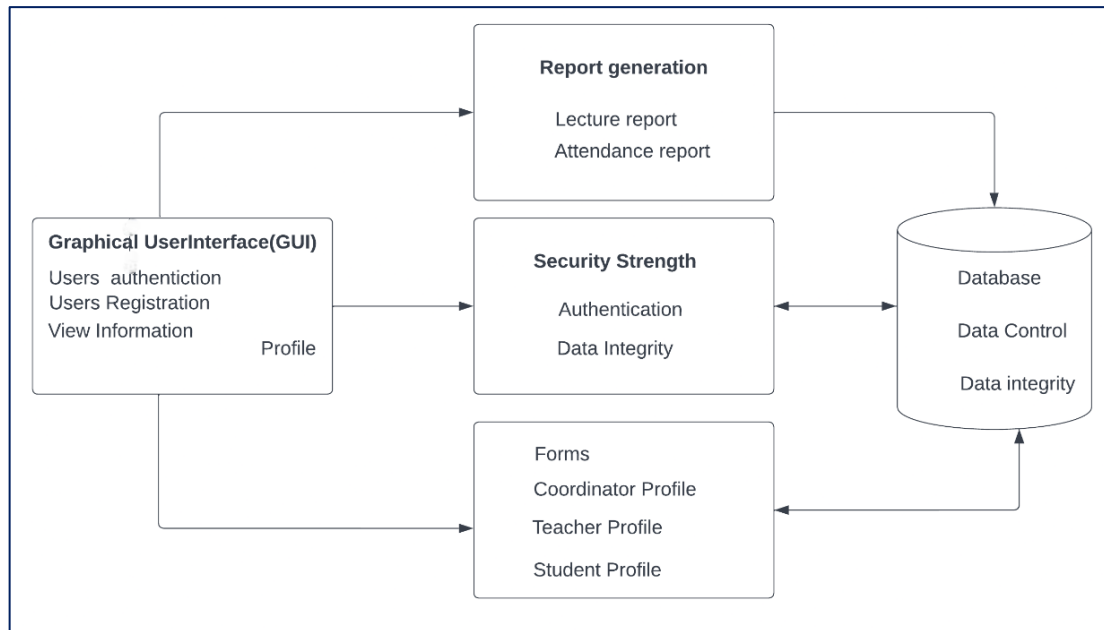


Fig.3.2 System Architecture

3.3.3 ER Diagram

ER diagram is a graphical representation of entities and their relationship to each other, typically used in computing regarding the organization of data within database or information systems. Entity is a piece of data, object or concept which described which data should store. Relationship is how data is shared between entities.

Entity

Entities are represented by rectangle. An entity is an object or concept that has its existence in the real world. It includes all those things about which data is collected. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Attributes

Which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.

For example, an employee's social security number might be the employee's key attribute.

Entity Set

It is a set of entities of the same type that share the same properties, or attributes.

Process

A process shows a transformation or manipulation of data flows within the system.

Actions

Actions are represented by diamond shapes, show how two entities share information in the database.

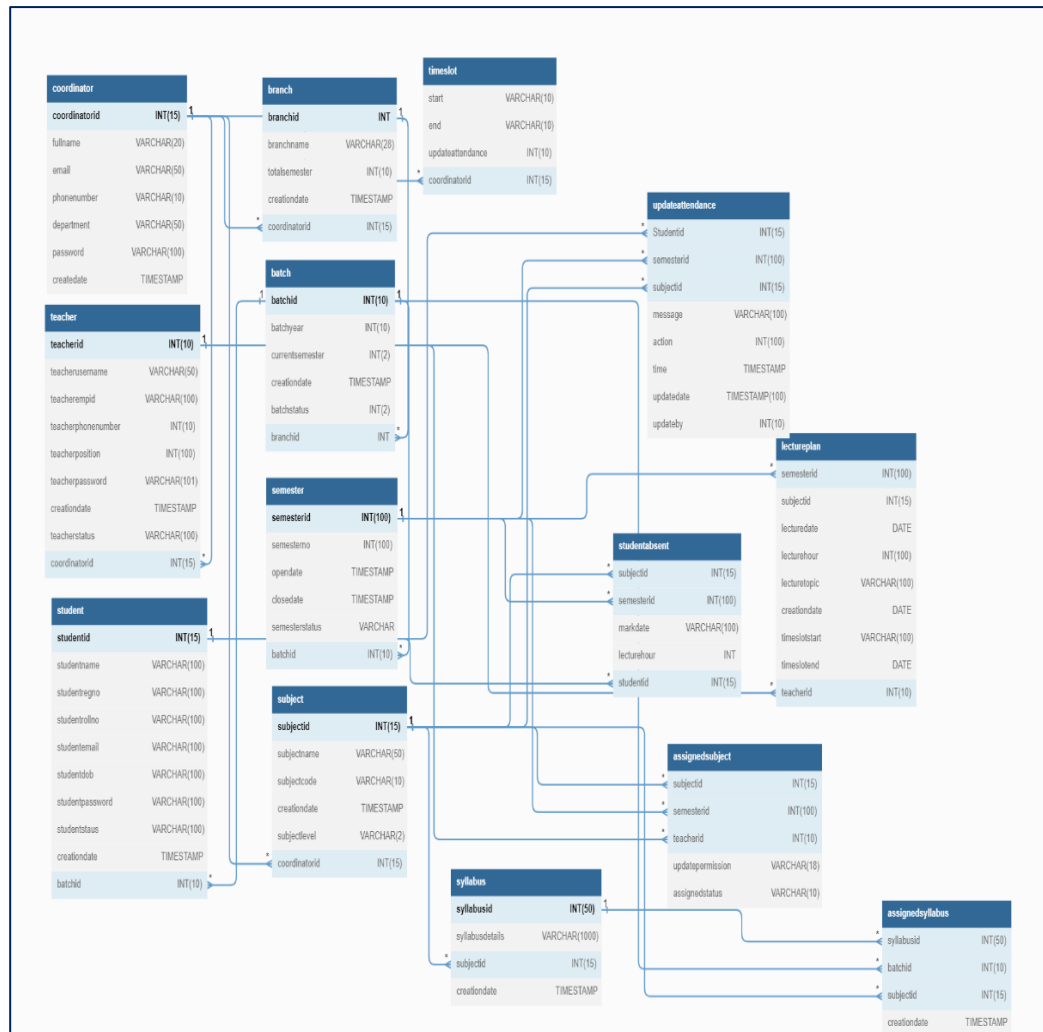


Fig.3.3 ER Diagram

3.3.4 DFD

A Data Flow Diagram (DFD) is a visual representation of the information flow within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who

plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

➤ **Level-0 DFD**

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

➤ **Level-1 DFD**

In level-1 DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

➤ **Level-2 DFD**

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

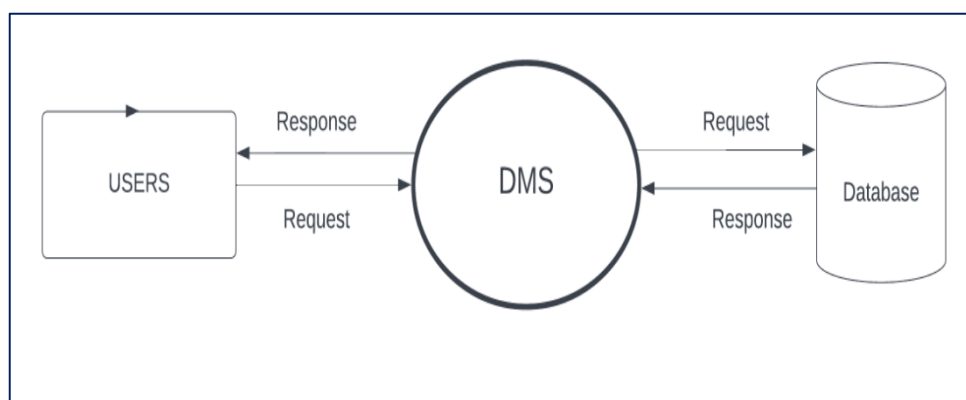


Fig.3.4 Level-0 DFD

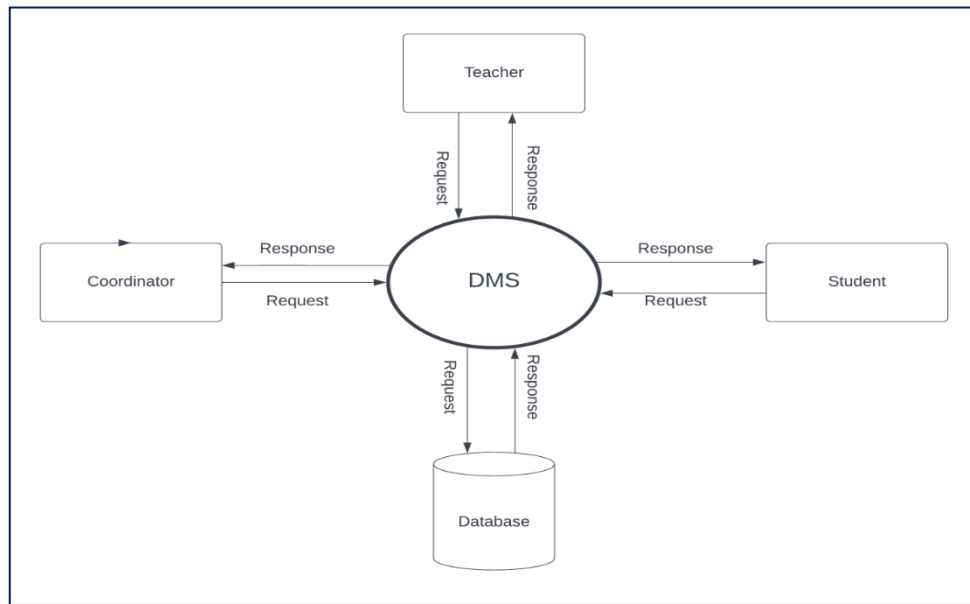


Fig.3.5 Level-1 DFD

3.3.5 Activity Diagrams

Activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent.

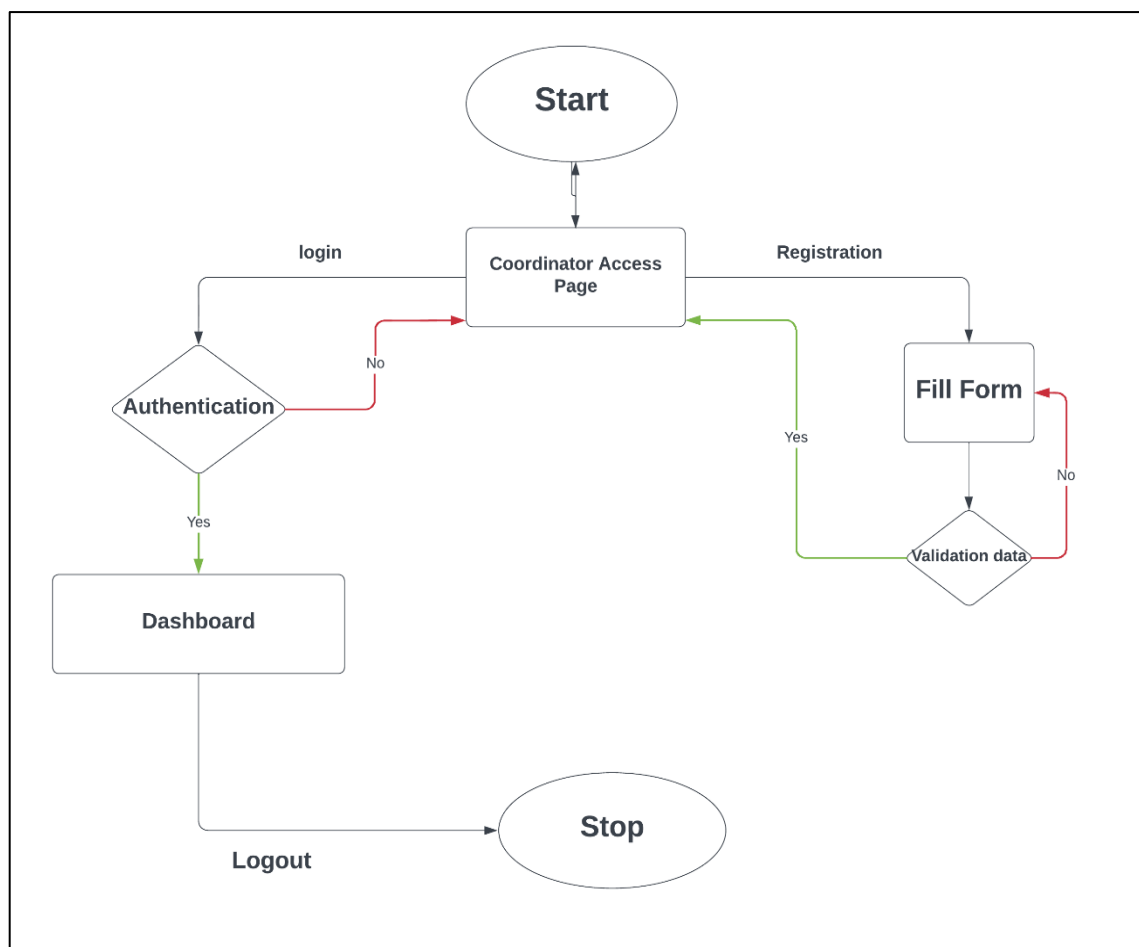


Fig.3.6 Activity Diagram of Coordinator Module

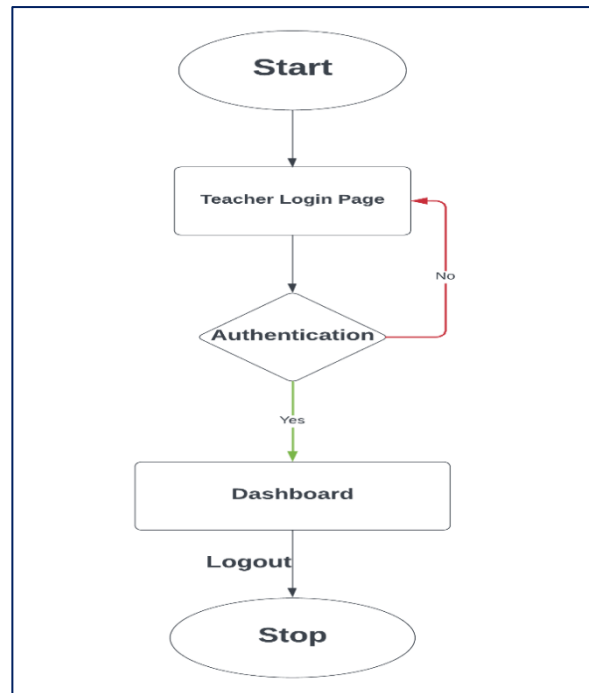


Fig.3.7 Activity Diagram of Teacher Module

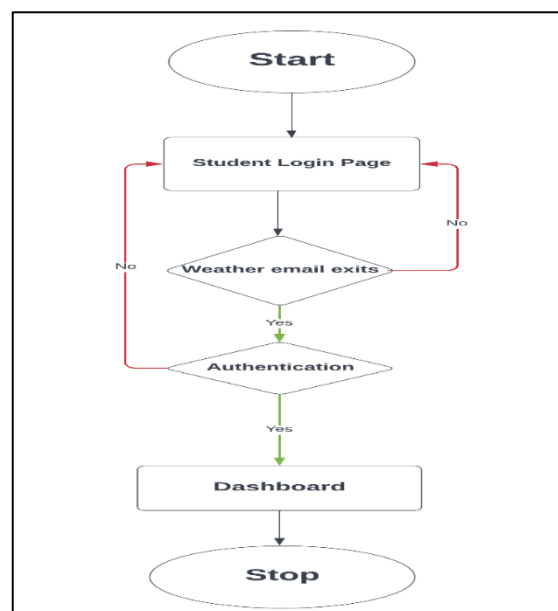


Fig.3.8 Activity Diagram of Student Module

CHAPTER 4

IMPLEMENTATION

A web application is an application program that is usually stored on a remote server, and users can access it through the use of Software known as web browser. It is a client-server application where a browser acts as the client and a web-server as the server. Web application logic is distributed between the client and the server, data storage is performed mainly on the server. Data is exchanged over the network using Hypertext Transfer Protocol (HTTP).

4.1 Client Server Architecture

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-server Model are email, World wide Web, etc.

The client-server architecture is also termed as a network-computing structure because every request and their associated services are distributed over a network. One special feature is that the server computer has the potential to manage numerous clients at the same time. Also, a single client can connect to numerous servers at a single timestamp, where each server provides a different set of services to that specific client.

- **Client:** When we talk the word Client, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a Client is a computer (Host) i.e. capable of receiving information or using a particular service from the service providers (Servers).
- **Servers:** Similarly, when we talk the word Servers, It mean a person or medium that serves something. Similarly in this digital world a Server is

a remote computer which provides information (data) or access to particular services.

4.1.1 Working of Client-Server Architecture

In this section we are going to have a look at how the Internet works via, web browsers. So, its basically the Client requesting something and the Server serving it as long as its present in the database.

Following are few steps by which a client interacts with servers:

- 1) User enters the URL of the website or file. The Browser then requests the DNS Server.
- 2) DNS Server lookup for the address of the WEB Server.
- 3) DNS Server responds with the IP address of the WEB Server.
- 4) Browser sends over an HTTP/HTTPS request to WEB Server's IP (provided by DNS server).
- 5) Server sends over the necessary files of the website.
- 6) Browser then renders the files and the website is displayed. This rendering is done with the help of DOM interpreter, CSS interpreter and JS Engine collectively known as the JIT Compilers.

4.1.2 Advantages of Client-Server model

- Centralized system with all data in a single place.
- Cost efficient requires less maintenance cost and Data recovery is possible.
- The capacity of the Client and Servers can be changed separately.

4.1.3 Disadvantages of Client-Server model

- Clients are prone to viruses, Trojans and worms if present in the Server or uploaded into the Server.
- Server are prone to Denial of Service (DOS) attacks.

- Data packets may be spoofed or modified during transmission.
- Phishing or capturing login credentials or other useful information of the user are common and MITM(Man in the Middle) attacks are common.

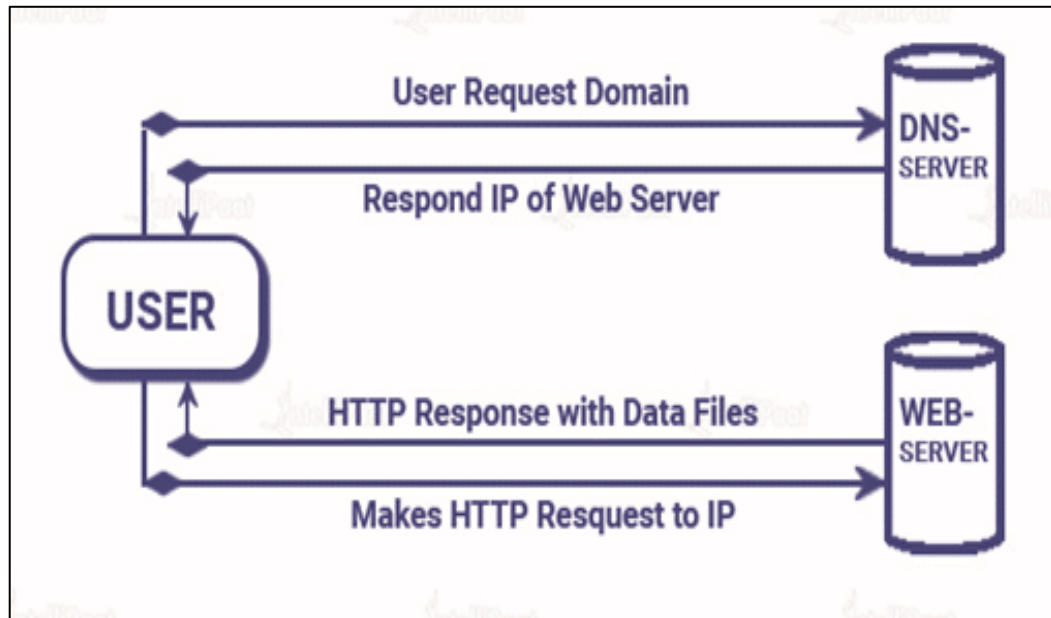


Fig.4.1: Client Server Architecture

4.2 HTTP

The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load web pages using hypertext links. HTTP is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack. A typical flow over HTTP involves a client machine making a request to a server, which then sends a response message. An HTTP request is the way internet communications platforms such as web browsers ask for the information they need to load a website. Each HTTP request made across

the Internet carries with it a series of encoded data that carries different types of information. A typical HTTP request contains:

1. HTTP version type
2. URL
3. HTTP method
4. HTTP request headers
5. Optional HTTP body

➤ **HTTP method**

An HTTP method, sometimes referred to as an HTTP verb, indicates the action that the HTTP request expects from the queried server. For example, two of the most common HTTP methods are ‘GET’ and ‘POST’. A ‘GET’ request expects information back in return (usually in the form of a website), while a ‘POST’ request typically indicates that the client is submitting information to the web server (such as form information, e.g. a submitted username and password).

➤ **HTTP request headers**

HTTP headers contain text information stored in key-value pairs, and they are included in every HTTP request (and response, more on that later). These headers communicate core information, such as what browser the client is using, what data is being requested.

➤ **Request body**

The body of a request is the part that contains the ‘body’ of information the request is transferring. The body of an HTTP request contains any information being submitted to the web server, such as a username and password, or any other data entered into a form.

➤ **HTTP response**

An HTTP response is what web clients (often browsers) receive from an Internet server in answer to an HTTP request. These responses communicate valuable information based on what was asked for in the HTTP request.

4.3 HTTPS

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS simply uses an encryption protocol over HTTP in order to increase the security of communication. The protocol is called Transport Layer Security (TLS), although formerly it was known as Secure Sockets Layer (SSL). This protocol secures communications by using what's known as an asymmetric public key infrastructure. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider. Any website, especially those that require login credentials, should use HTTPS.

In modern web browsers such as Chrome, websites that do not use HTTPS are marked differently than those that are. When a user connects to a webpage, the webpage will send over its SSL certificate which contains the public key necessary to start the secure session. The two computers, the client and the server, then go through a process called an SSL/TLS handshake, which is a series of back-and-forth communications used to establish a secure connection.

4.4 Web Development Technologies

Web development technologies refer to the multitude of programming languages and tools that are used to produce dynamic and fully-featured websites and applications.

4.5 Front-end (client-side) technologies

Front-end technologies are for the “client side” of your website or application. They’re used to develop the interactive components of your website, and produce the elements that users see and interact with. This includes text colours and styles, images, buttons and navigation menus.

Following are the front-end technologies that we used in the development of our web application:

4.5.1 HTML

HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications.

Hypertext: Hypertext simply means "Text within Text." A text THAT has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hypertext is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

HTML5 is the newest version of Hypertext Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG (Web Hypertext Application Technology Working Group)

which are involved in the development of HTML 5 version, and still, it is under development.

4.5.2 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document.

List of frameworks used:

1. Bootstrap.css
2. SASS.css

4.5.3 JavaScript

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other

graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses JS to provide several forms of interactivity and simplicity.

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

List of frameworks used:

1. JQuery.js
2. SweetAlert.js
3. FontAwesome.js

4.6 Back-end (server-side) technologies

Back-end technologies are for the “server side” of your website or application. They’re for developing the technical foundation. They store and arrange data and make sure everything on the front-end works.

Following are the back-end technologies that we used in the development of our web application:

4.6.1 PHP Object Oriented

Hypertext Pre-processor (or simply PHP) is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Pre-processor*.

PHP is well suited for web development. Therefore, it is used to develop web applications. PHP code can be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks.

PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

List of frameworks used

1. PDO (PHP Data Object)
2. PHP Excel

4.6.2 AJAX

AJAX is an acronym for Asynchronous JavaScript and XML. It is a group of inter-related technologies like JavaScript, DOM, XML, HTML/XHTML, CSS, XMLHttpRequest etc. AJAX allows you to send and receive data asynchronously without reloading the web page. So, it is fast.

AJAX allows you to send only important information to the server not the entire page. So only valuable data from the client side is routed to the server side. It makes your application interactive and faster.

4.6.3 SQL

SQL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc. SQL is not a database system, but it is a query language. This database language is mainly designed for maintaining the data in relational database management systems as it is the standard language for relational database systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). All the Relational Database Management Systems (RDBMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

4.7 Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data. SQL or Structured Query Language is used to operate on the data stored in a database.

4.7.1 MySQL

MySQL is the most popular database management system software used for managing the relational database. It is based on SQL. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

4.8 XAMPP

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

CHAPTER 5

SYSTEM TESTING

Developed system is tested against the user requirements, which are gathered at the beginning. Actual data from the client's environment is used. The chapter also includes aspects such as test plan, test cases, use of test automation tools and testing frameworks

The following items will be considered in testing:

5.1 Co-ordinator Test Case

- Co-ordinator Login
- Co-ordinator Registration
- Co-ordinator Dashboard activities(Adding new program, batch, subjects, syllabus, students, teachers, etc)
- Co-ordinator settings(Time slot, update time, attendance request time, etc)
- Co-ordinator Logout

5.2 Teacher Test Case

- Teacher Login
- Teacher Dashboard Activities(Maintaining attendance and lecture plan details)
- Teacher Logout

5.3 Student Test Case

- Student Authentication
- Student Login
- Student Profile Updation
- Student Dashboard Activities(Attendance details,lecture plan,syllabus, etc)
- Student Logout

5.4 PDF Generation Test Case

- Subject wise
- Semester wise
- Lecture plan

Case	Input Data	Expected Results
Login page	Correct User Name Correct Password	Displays the dashboard of the user.
	Correct User Name Incorrect Password	Displays error message
	Incorrect User Name Correct Password	Displays error message.
Registration Page	Complete and Correct Information	Displays success message for registration.
	Incomplete and Correct Information	Displays error message for registration.
	Incomplete and Incorrect Information	Displays error message for registration.
Dashboard Activities	Correct Information Correct Operations (Insert, Delete, Update, Read)	Displays success message and successful updating of system.
	Correct Information Incorrect Operations (Insert, Delete, Update, Read)	Displays error message and no change in initial state of system.
	Incorrect Information Incorrect Operations (Insert, Delete, Update, Read)	Displays error message and no change in initial state of system.
Logout menu	User click the logout menu	Redirect to the login page All sessions get closed.

Table 5.1: General Testing Cases

Case	Input Data	Expected Results
PDF Generation	Select Subject from active semesters.	PDF generated with attendance of all enrolled students in that subject with calculated percentage.
	Select semester from current batch.	PDF generated with attendance of all enrolled students in all subjects in the particular semester.
	Select lecture plan of subject.	PDF generated with details of all lectures in that subject.

Table 5.2: PDF Generation Testing Cases

CHAPTER 6

CODING SAMPLES

6.1 Co-ordinator registration

```
<?php
include_once('db_connection.php');
class registrations extends db_connection{
protected $email;
protected $username;
protected $phonenummer;
protected $department;
protected $password;
protected $confirmpassword;

function __construct
($username, $email, $phonenummer, $department, $password, $confirmpassword) {
$this->username = trim(htmlspecialchars($username));
$this->email = trim(htmlspecialchars($email));
$this->phonenummer = trim(htmlspecialchars($phonenummer));
$this->department = trim(htmlspecialchars($department));
$this->password = trim(htmlspecialchars($password));
$this->confirmpassword = trim(htmlspecialchars($confirmpassword));
parent::__construct();

if ($this->validate_data()){
if ($this->checkEmail()){
$this->registerEmail();
}
else{
echo 2;
}
}
else{
echo 1;
}
}

private function validate_data(){
if (!preg_match("/^[a-zA-Z-0-9' ]*$/", $this->username)) {
return false;
}
if (!preg_match("/^[a-zA-Z-' ]*$/", $this->department)) {
return false;
}
if (!filter_var($this->email, FILTER_VALIDATE_EMAIL)) {
return false;
}
if (!preg_match('/^[0-9]*$/', $this->phonenummer)) {
return false;
}
}
```

```

if(strlen($this->phonenumber) != 10 ){
return false;
}
if (strlen($this->password) < 8 || $this->password != $this-
>confirmpassword){
return false;
}
return true;
}

private function checkEmail(){

$sql = $this->conn->prepare("SELECT * FROM coordinator WHERE email = ?");
$sql->bindParam(1, $this->email);
$sql->execute();
if($sql->rowCount() > 0){
return false;
}
else{
return true;
}
}

private function registerEmail(){
$newpassword = password_hash($this->password, PASSWORD_DEFAULT);
try{
$sql = $this->conn->prepare("INSERT INTO `coordinator` (`fullName`,
`email`, `phonenumber`, `department`, `password`, `createdate`) VALUES
(?,?,?,?,?,current_timestamp())");
$sql->bindParam(1, $this->username);
$sql->bindParam(2, $this->email);
$sql->bindParam(3, $this->phonenumber);
$sql->bindParam(4, $this->department);
$sql->bindParam(5, $newpassword);

if ($sql->execute()){
echo 3;
}
else{
// echo "User registrations failed";
echo 0;
}
}catch(PDOException $obj1){

echo 0;
}
}
}
}

```

```

if (isset($_POST['email_']) && isset($_POST['connecti on'])) {
    $run = new
    registrations($_POST['user_name'], $_POST['email_'], $_POST["phone_number"], $_
    _POST["depart_ment"], $_POST["pass_word"], $_POST["password_2"]);

    $run->closeConnecti on();
} else {
    header("Locati on: ../coordi natorl ogi n_si gnup. html ");
}
?>

```

6.2 Co-ordinator login

```

<?php
require_once("db_connecti on. php");
class login extends db_connection {
    protected $email;
    protected $password;
    function __construct($email , $password) {
        parent::__construct();
        $this->email = trim(html speci al chars($email));
        $this->password = trim(html speci al chars($password));
        if($this->val idate()) {
            if($this->checkEmail Exi sts()) {
                // echo 3;
            } else {
                echo 1;
                // email or password wrong
            }
        } else {
            echo 0;
            // input val idate
        }
    }
    private function val idate() {
        if (!filter_var($this->email, FILTER_VAL IDATE_EMAIL)) {
            return false;
        }
        return true;
    }
    private function checkEmail Exi sts() {
        $sql = $this->conn->prepare("SELECT * FROM coordi nator WHERE email = ?");
        $sql->bindParam(1, $this->email);
        $sql->execute();
        if($sql->rowCount() > 0) {
            $resul t = $sql->fetch(PDO:: FETCH_ASSOC);

```

```

$pass = $result['password'];
if(password_verify($this->password, $pass)){
    session_start();
    $_SESSION['active'] = true;
    $_SESSION['userid'] = $result['coordinatorid'];
    $_SESSION['username'] = $result['fullname'];
    echo 3;
    exit();

    return true;
}else{

    return false;
}
}else{
    return false;
}

}
}

if (isset($_POST['emaillogin']) && isset($_POST['connection'])){
    $run = new Login($_POST['emaillogin'], $_POST['passwordlogin']);

    $run->closeConnection();
}else{
    header("Location: ../coordinatorlogin_signup.html");
}

?>

```

6.3 JS code for setting limit of days for attendance

```

function UpdateDays(days) {
$.ajax({
url: "../coordinator/modal/sendmodaldata/senddays.php",
type: "POST",
beforeSend: function() {

$("#update_days").html("wait..");
},
data: {
getdays: days,
connection: true
},
success: function(data) {

```



```
if (data == 3) {
    swal ("Good Job!", "Day Updated", "success");
    $("#update_days").html ("Update Days");

} else if (data == 1) {
    swal ("ohoho ",
    "Couln' t Update Days...!",
    "error");
    $("#update_days").html ("Update Days");

} else {
    swal ("ohoho ",
    "Something went wrong...!",
    "error");
    $("#update_days").html ("Update Days");
}
}
});
}
```

6.4 Adding a program/branch

```
<?php
require_once("../.. /inner/db_connection. php");
class sendBranch extends db_connection{
    private $get_Branch;
    private $get_Semester;
    private $get_Coordinatorid;

    function __construct($getBranch, $getSemester, $getCoordinatorid){
        parent::__construct();
        $this->get_Branch = trim(html special chars($getBranch));
        $this->get_Semester = trim(html special chars($getSemester));
        $this->get_Coordinatorid = trim(html special chars($getCoordinatorid));

        require_once("../.. /.. /coordinator/checkDataExists/coordinator. php");
        if ($this->checkValid()){
            //data is valid
            if ($countCoordinator > 0){
                $this->sendData();
            }else{
                echo 0;
            }
        }else{
            //data is not valid to send into database
            echo 1;
        }
    }
}
```

```

}
}

private function checkValid(){
if($this->get_Semester > 13 || !is_numeric($this->get_Semester) ||
!preg_match("/^[a-zA-Z-, -.' ]*$/", $this->get_Branch) ){
return false;
}

return true;
}

private function sendData(){
$sql = $this->conn->prepare("SELECT * FROM `branch` WHERE `coordinatorid`
= ? && `branchname` = ? ");
$sql->bindParam(1, $this->get_Coordinatorid);
$sql->bindParam(2, $this->get_Branch);
$sql->execute();
if ( $sql->rowCount() > 0 ){
// name already present in the branch
echo 2;
}else{
$sql = $this->conn->prepare("INSERT INTO `branch` (`branchname`,
`total semester`, `coordinatorid`, `creationdate`) VALUES
(?,?,?,current_timestamp())");
$sql->bindParam(1, $this->get_Branch);
$sql->bindParam(2, $this->get_Semester);
$sql->bindParam(3, $this->get_Coordinatorid);
if( $sql->execute()){
//data send
echo 3;
}else{
//data not send
echo 1;
}
}
}

}
}

if(isset($_POST["get_Coordinator"]) && isset($_POST['connection']) &&
isset($_POST['get_Branch'])){
$run = new
sendBranch($_POST['get_Branch'], $_POST['get_Semester'], $_POST["get_Coordin
ator"]);
$run->closeConnection();
}else{
header("Location: ../.../coordinatorlogin_signup.html");
}

```

```
}  
?>
```

6.5 Delete a subject

```
<?php  
include_once("../inner/db_connection.php");  
class deletesubject extends db_connection  
{  
function __construct($getsubjectid)  
{  
parent::__construct();  
$check = $this->conn->prepare("SELECT * FROM `subject` WHERE `subjectid`  
= ?");  
$check->bindParam(1, $getsubjectid);  
$check->execute();  
$count = $check->rowCount();  
  
if ($count > 0) {  
$sqlpre = $this->conn->prepare("SELECT * FROM `subject` INNER join  
lectureplan on subject.subjectid = lectureplan.subjectid WHERE  
subject.subjectid = ?");  
$sqlpre->bindParam(1, $getsubjectid);  
$sqlpre->execute();  
$count = $sqlpre->rowCount();  
if ($count == 0) {  
$sql = $this->conn->prepare("DELETE FROM `assignedsubject` WHERE  
`subjectid` = ?");  
$sql->bindParam(1, $getsubjectid);  
if ($sql->execute()) {  
$sql = $this->conn->prepare("DELETE FROM `subject` WHERE `subjectid` =  
?");  
$sql->bindParam(1, $getsubjectid);  
if ($sql->execute()) {  
echo 3;  
} else {  
echo 1;  
}  
} else {  
echo 1;  
}  
} else {  
echo 2;  
}  
} else {  
echo 1;  
}
```

```
}
}
}
if (isset($_POST['get_Subjectid']) && isset($_POST['connection'])) {
    $run = new deletesubject($_POST['get_Subjectid']);

    $run->closeConnection();
} else {
    header("Location: ../coordinatorlogin_signup.html");
}
```

6.6 Generate pdf of attendance for all subjects

```
<?php
session_start();
if (!isset($_SESSION['active']) || $_SESSION['active'] != true) {
    header("Location: ../../DMS/coordinator/coordinatorlogin_signup.html");
    die();
}

require_once("../fpdf.php");
class myPDF extends FPDF
{

    function header()

    {
        $this->Image('../Logo.png', 15, 10);
        $this->SetFont("Arial", 'B', 14);
        $this->Cell(276, 5, 'Department of Computer Science and Engineering', 0,
        0, 'C');
        $this->Ln();
        $this->SetFont('Times', '', 12);
        $this->Cell(276, 10, 'North Campus, University of
        Kashmir, Delina, Baramulla', 0, 0, 'C');
        $this->Ln();
        $this->SetFont('Times', '', 14);
        $this->Cell(276, 10, 'ATTENDANCE STATUS', 0, 0, 'C');
        $this->Ln(15);
    }
    function headerTable($year, $semno)
    {
        $this->SetTextColor(0, 0, 0);
        $this->SetFont('Times', '', 10);
        $this->Cell(40, 5, 'Batch/Semester: ' . $year . '/' . $semno, 0, 0, 'C');
        $this->SetFont('Times', '', 10);
        $this->Cell(200, 5, 'Minimum Attendance Required = 75%', 0, 0, 'C');
```

```

$this->Cell(23, 5, 'Session: ' . date("Y"), 0, 0, 'C');
$this->Ln();
$this->SetFont('Times', '', 10);
$date = date("d-m-Y");
$this->Cell(32, 5, 'Date: ' . $date, 0, 0, 'C');

$this->Ln(5);
$this->SetFont('Times', 'B', 12);
$this->Cell(40, 10, 'Enrollment NO', 1, 0, 'C');
}
function createHeader($subjectname)
{

$this->Cell(50, 10, $subjectname, 1, 0, 'C');
}
function createline()
{
$this->Ln();
}

function Footer()
{
$this->SetTextcolor(0, 0, 0);
$this->SetY(-15);
$this->SetFont('Times', '', 10);
$this->Cell(500, 10, 'Coordinator', 0, 0, 'C');
$this->Ln(2);
$this->SetFont('Arial', 'I', 8);

$this->Cell(0, 10, 'Page ' . $this->PageNo() . '/{nb}', 0, 0, 'C');
}

function viewTable($studentname)
{
$this->Cell(40, 10, $studentname, 1, 0, 'C');
}
function viewSubject($percentage)
{
$this->Cell(50, 10, $percentage, 1, 0, 'C');
}
}

require("../coordinator/dbcon.php");
if (isset($_POST['subjectlecturepdf']) && $_POST['selectsemesterno'] != 0
&& $_POST['select_batch_id'] != 0) {
$semesterid = $_POST['selectsemesterno'];
$batchid = $_POST['select_batch_id'];
$subjectLevel = $_POST['subjectlevel'];

```

```

if ($subjectLevel == 'T' || $subjectLevel == 'L') {

$pdf = new myPDF();
$pdf->AliasNbPages();
$pdf->AddPage('L', 'A4', 0);
$pdf->headerTable("2019", "4");
$sql2 = $conn->prepare("SELECT * FROM `subject` INNER JOIN
`assignedsubject` ON subject.subjectid = assignedsubject.subjectid INNER
JOIN `teacher` ON teacher.teacherid = assignedsubject.teacherid WHERE
assignedsubject.semesterid = ? && assignedsubject.assignedstatus !=
'disabled' && subject.subjectlevel = ?");
$sql2->bindParam(1, $semesterid);
$sql2->bindParam(2, $subjectLevel);
$sql2->execute();
$subjects = $sql2->fetchAll();
foreach ($subjects as $subjects_row) {
$pdf->createHeader($subjects_row['subjectname']);
}
$pdf->createLine();
if ($subjectLevel == "T") {
$sql1 = $conn->prepare("SELECT * FROM student where batchid = ?");
$sql1->bindParam(1, $batchid);
$sql1->execute();
$student_rolino = $sql1->fetchAll();
foreach ($student_rolino as $rolino) {
$pdf->viewTable($rolino['studentrolino']);
foreach ($subjects as $subjects_row) {

$totalclasssql = $conn->prepare("SELECT * FROM `lectureplan` WHERE
`subjectid` = ? && `semesterid` = ?");
$totalclasssql->bindParam(1, $subjects_row['subjectid']);
$totalclasssql->bindParam(2, $semesterid);
$totalclasssql->execute();
$fetchclass = $totalclasssql->fetchAll(PDO::FETCH_ASSOC);
$totalclass = 0;

foreach ($fetchclass as $countclass) {

$totalclass = $totalclass + $countclass['lecturehour'];
}

$findabsent = $conn->prepare("SELECT * FROM `studentabsent` WHERE
`studentid` = ? && `subjectid` = ? && `semesterid` = ?");
$findabsent->bindParam(1, $rolino['studentid']);

```

```

$findabsent->bindParam(2, $subjects_row['subjectid']);
$findabsent->bindParam(3, $semesterid);
$findabsent->execute();
$fetchasbsentcount = $findabsent->fetchAll(PDO::FETCH_ASSOC);
$absentcount = 0;
foreach ($fetchasbsentcount as $somecount) {
$absentcount = $absentcount + $somecount['lecturehour'];
}
$presentcount = $totalclass - $absentcount;
if ($totalclass == 0) {
$percentage = 0;
} else {
$percentage = ceil($presentcount / $totalclass * 100);
}
$pdf->viewSubject($percentage);
}
$pdf->createline();
}
} else {
$group = array("G1", "G2");
$groups = "BOTH";
for ($i = 0; $i <= 1; $i++) {
$sql1 = $conn->prepare("SELECT * FROM student where batchid = ? &&
group_id = ?");
$sql1->bindParam(1, $batchid);
$sql1->bindParam(2, $group[$i]);
$sql1->execute();
$student_rolino = $sql1->fetchAll();
foreach ($student_rolino as $rolino) {
$pdf->viewTable($rolino['studentrolino']);
foreach ($subjects as $subjects_row) {
$totalclasssql = $conn->prepare("SELECT * FROM `lectureplan` WHERE
`subjectid` = ? && `semesterid` = ? && groups = ? UNION SELECT * FROM
`lectureplan` WHERE `subjectid` = ? && `semesterid` = ? && groups = ?");
$totalclasssql->bindParam(1, $subjects_row['subjectid']);
$totalclasssql->bindParam(2, $semesterid);
$totalclasssql->bindParam(3, $group[$i]);
$totalclasssql->bindParam(4, $subjects_row['subjectid']);
$totalclasssql->bindParam(5, $semesterid);
$totalclasssql->bindParam(6, $groups);
$totalclasssql->execute();
$fetchclass = $totalclasssql->fetchAll(PDO::FETCH_ASSOC);
$totalclass = 0;
foreach ($fetchclass as $countclass) {

$totalclass = $totalclass + $countclass['lecturehour'];
}
}
}

```

```

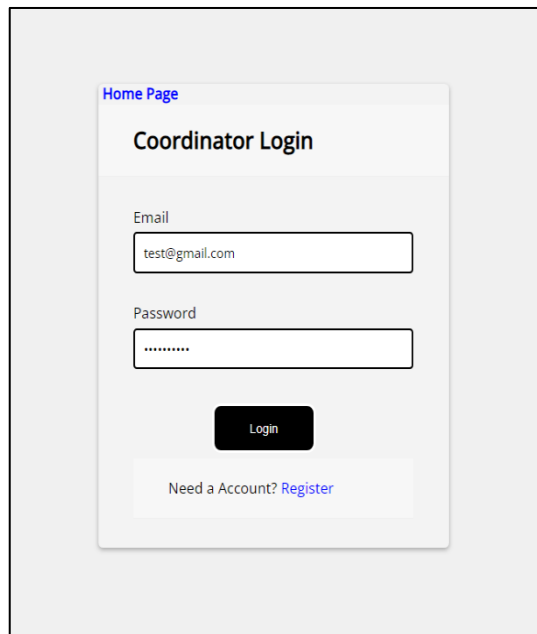
$findabsent = $conn->prepare("SELECT * FROM `studentabsent` WHERE
`studentid` = ? && `subjectid` = ? && `semesterid` = ?");
$findabsent->bindParam(1, $rollno['studentid']);
$findabsent->bindParam(2, $subjects_row['subjectid']);
$findabsent->bindParam(3, $semesterid);
$findabsent->execute();
$fetchasbsentcount = $findabsent->fetchAll(PDO::FETCH_ASSOC);
$absentcount = 0;
foreach ($fetchasbsentcount as $somecount) {
$absentcount = $absentcount + $somecount['lecturehour'];
}
$presentcount = $totalclass - $absentcount;
if ($totalclass == 0) {
$percentage = 0;
} else {
$percentage = ceil($presentcount / $totalclass * 100);
}
$pdf->viewSubject($percentage);
}
$pdf->createline();
}
}
}
$pdf->Output();
} else {
header("Location: .../.../coordinator/dashboard.php");
$conn = null;
die();
}
} else {
header("Location: .../.../coordinator/dashboard.php");
$conn = null;
die();
}

$conn = null;

```


CHAPTER 7

SYSTEM INTERFACES



Home Page

Coordinator Login

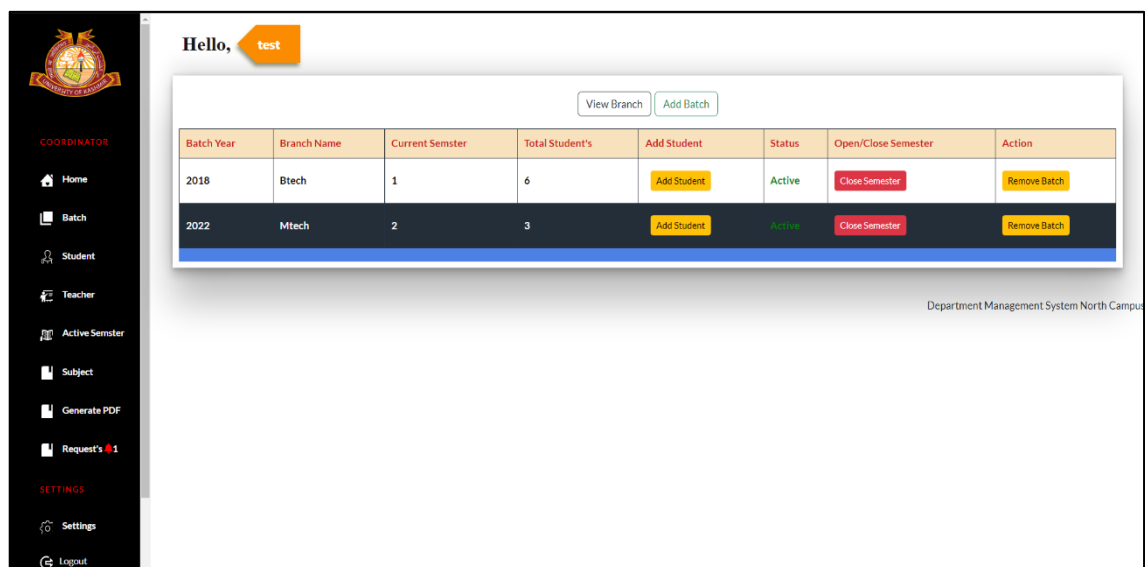
Email

Password

Login

Need a Account? [Register](#)

Fig.7.1: Co-ordinator login page



Hello, **test**

[View Branch](#) [Add Batch](#)

Batch Year	Branch Name	Current Semester	Total Student's	Add Student	Status	Open/Close Semester	Action
2018	Btech	1	6	Add Student	Active	Close Semester	Remove Batch
2022	Mtech	2	3	Add Student	Active	Close Semester	Remove Batch

Department Management System North Campus

COORDINATOR

- Home
- Batch
- Student
- Teacher
- Active Semester
- Subject
- Generate PDF
- Request's **1**

SETTINGS

- Settings
- Logout

Fig.7.2: Dashboard page with menus

Hello, **test**

Active Semester

Batch Year	Branch Name	Current Semester	Total Student's	Starting Date	Closing Date	Action
2018	Btech	1	901066	2022-12-08 21:40:12		Subject assign
2022	Mtech	2	901068	2022-12-08 22:01:45		Subject assign

Closed Semester

Batch Year	Branch Name	Semester No	Total Student's	Starting Date	Closing Date	Action
2022	Mtech	1	901067	2022-12-08 22:01:42	2022-12-08 22:01:45	More Information

Department Management System North Campus

Fig.7.3: Active and Closed semesters

Hello, **test**

→PDF Subject Record

→PDF LECTURE

↓PDF Semester

SELECT A SEMESTER TO GENERATE PDF

Chosen a batch

Btech2018

Chosen Semester No

1

Chosen Subject Mode

Theory Subject's

[Generate pdf](#)

Fig.7.4: PDF generation for a particular subject

Hello, test

↓ View Student

Select the Batch to Show Student's Enrolled

2018 - Btech

S.No	Student Email	Student Name	Student Enrollment	Group	Status	Action
1	saqlain@gmail.com	Saqlain Khan	18040681120	G2	inactive	Remove More Information
2	ubaid@gmail.com	Dar Ubaid Nazir	18048112001	G1	inactive	Remove More Information
3	parvaiz@gmail.com	Parvaiz Bhat	18048112002	G1	inactive	Remove More Information
4	irshad23@gmail.com	Irshaid wani	18048112003	G1	inactive	Remove More Information
5	izhar11@gmail.com	Izhar ul hak	18048112004	G2	inactive	Remove More Information
6	Zamin1@gmail.com	Zamin ahmad dar	18048112005	G2	inactive	Remove More Information

1

→ Extra Attendance

→ Update Attendance

Fig.7.5: Enrolled students in a batch

Hello, test

ATTENDANCE UPDATE REQUEST FORM

S.No	Teacher Name	Subject Name		Action
1	test_teacher	networking	Grant	Cancel

Department Management System North Campus

Fig.7.6: Attendance update request form

Syllabus View Information

JAVA Programming - CSE-7122L

Detailed Syllabus

INTRODUCTION TO JAVA Programming

Assigned syllabus to batch year

2018

Assign syllabus to a batch

Select a Batch

Btech - 2018

Assign

1

Close


Fig.7.7: Syllabus details

Hello, test

Add Subject

S.No	Subject Id	Subject Name	Subject Code	Subject Level	Syllabus	Action
1	901072	networking	cse-7122	Theory	Add Syllabus View Syllabus	Edit Remove More Information
2	901073	c programming lab	cse-1130L	Lab	Add Syllabus View Syllabus	Edit Remove More Information
3	901074	Machine Learning	CSE-1475	Theory	Add Syllabus View Syllabus	Edit Remove More Information
4	901075	JAVA Programming	CSE-7122L	Lab	Add Syllabus View Syllabus	Edit Remove More Information
5	901076	Compiler Design	cse-1126	Theory	Add Syllabus View Syllabus	Edit Remove More Information
6	901077	Database	CSE-1122L	Theory	Add Syllabus View Syllabus	Edit Remove More Information
7	901078	Microprocessor	CSE-7878	Theory	Add Syllabus View Syllabus	Edit Remove More Information
8	901079	Microprocessor Lab	CSE-7878L	Lab	Add Syllabus View Syllabus	Edit Remove More Information

Fig.7.8: List of subjects



Department of Computer Science and Engineering
North Campus, University of Kashmir, Delina, Baramulla

ATTENDANCE STATUS

Batch/Semester: 2019/4 Minimum Attendance Required = 75% Session: 2022
Date: 09-12-2022

Enrollment NO	networking	Machine Learning	Database	Microprocessor	Compiler Design
18048112001	100	0	0	0	0
18048112002	100	0	0	0	0
18048112003	75	0	0	0	0
18048112004	75	0	0	0	0
18048112005	100	0	0	0	0
18040681120	50	0	0	0	0

Coordinator

Page 1/1

Fig.7.9: PDF of attendance details

↓ Add Group's

↓ Add Time slot

Select a Batch to assign group's

Select a Batch

Enter Maximum days to mark attendance from current date.

5

Update Days

SELECT TIME SLOTS

Start Time	End Time	Update Day's
09:30	16:00	5

Select a Start Time

Select a End Time

Set Time

Fig.7.10: Add group and set working time

Subject Assign Semester Information

SELECT A SUBJECT TO BE ASSIGNED TO THIS SEMESTER

Open this to Select a Subject

Open this to Select a Teacher

Assign

SUBJECT ASSIGNED TO THIS SEMESTER

S.No	Subject Name	Subject Code	Teacher Assigned	Action
1	networking	cse-1122	test_teacher	Remove
2	Machine Learning	CSE-1475	test_john	Remove
3	Database	CSE-1123	test_john	Remove
4	Microprocessor	CSE-7878	test_dart	Remove
5	Microprocessor Lab	CSE-7878	test_dart	Remove
6	c programming lab	cse-1130L	test_sam	Remove
7	web programming	CSE-1123	test_sam	Remove
8	Compiler Design	cse-1126	test_teacher	Remove

Fig.7.11: Subjects assigned to a semester

Subject Assigned to Teacher

SUBJECT ASSIGNED TO THIS TEACHER

S.No	Subject Name	Subject Code	Batch Year	Semester No
1	networking	cse-1122	2018	1
2	Compiler Design	cse-1126	2018	1

Close

Fig.7.12: Subjects assigned to a teacher

Hello, **test**

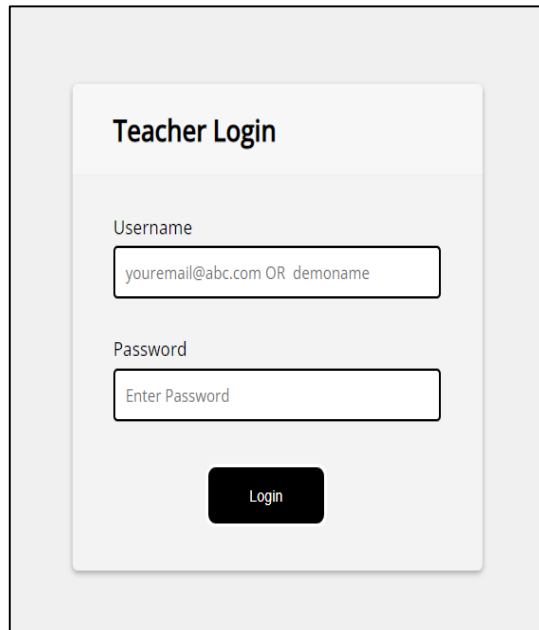
Add teacher

S.No	Teacher ID	Username	Emp ID	Phone Number	Position	Action
1	701059	test_teacher	123456	7894561230	Assistant Professor	Edit Remove More Information Subject assigned
2	701060	test_john	789654123	6547891288	Assistant Professor	Edit Remove More Information Subject assigned
3	701061	test_bob	21212	8585857423	Assistant Professor	Edit Remove More Information Subject assigned
4	701062	test_sam	123458	9636547852	Contractual	Edit Remove More Information Subject assigned
5	701063	test_dart	789542	5252415875	Contractual	Edit Remove More Information Subject assigned

1

Department Management System North Campus

Fig.7.13: Teachers list



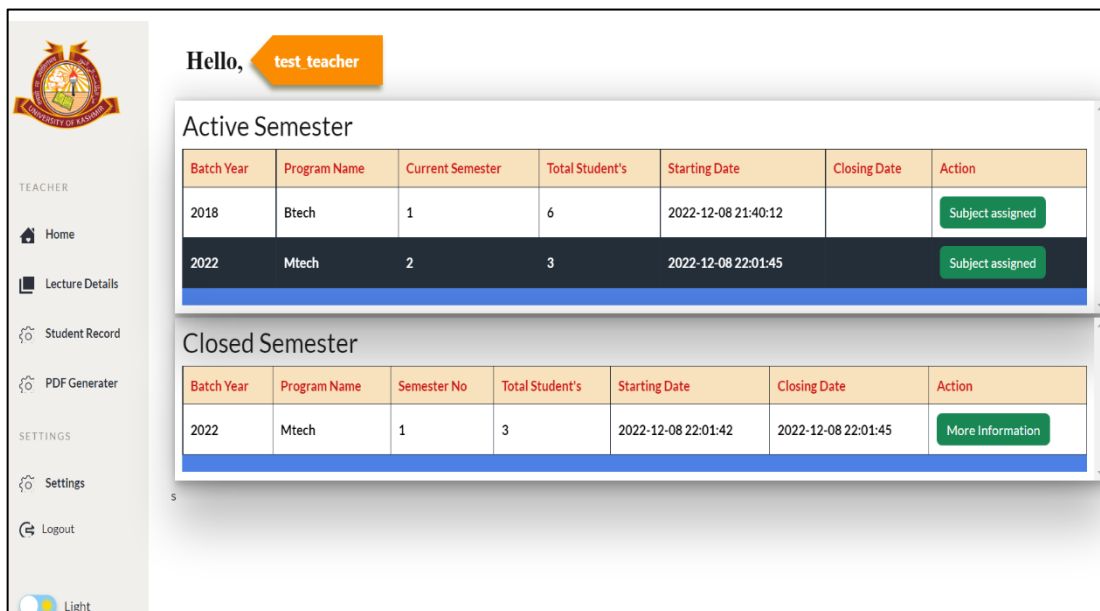
Teacher Login

Username
youremail@abc.com OR demoname

Password
Enter Password

Login

Fig.7.14:Teacher login page



Hello, test teacher

Active Semester

Batch Year	Program Name	Current Semester	Total Student's	Starting Date	Closing Date	Action
2018	Btech	1	6	2022-12-08 21:40:12		Subject assigned
2022	Mtech	2	3	2022-12-08 22:01:45		Subject assigned

Closed Semester

Batch Year	Program Name	Semester No	Total Student's	Starting Date	Closing Date	Action
2022	Mtech	1	3	2022-12-08 22:01:42	2022-12-08 22:01:45	More Information

TEACHER

- Home
- Lecture Details
- Student Record
- PDF Generator

SETTINGS

- Settings
- Logout

Light

Fig.7.15:Teacher dashboard page

Assigned Subject to Teacher

SUBJECT ASSIGNED TO THIS SEMESTER

S.No	Subject Name	Subject Code	Year/Sem No	Mode	Action
1	DataBase	CSE-1272	2018/1	Theory	Mark Attendance
2	Database Lab	CSE-1711L	2018/1	Lab	Mark Attendance
3	Compiler Design	CSE-1212	2018/1	Theory	Mark Attendance
4	Computer Networks	CSE-1712	2018/1	Theory	Mark Attendance
5	Computer Network Lab	CSE-1712L	2018/1	Lab	Mark Attendance

Close

Fig.7.16:List of subjects assigned to a teacher

networking

Enter Today's Lecture Topic

Introduction to Computer's

Select Total Hours

1

Select Today's Lecture Date

12/09/2022

Select Time Slot

09:50

Select a Mode

PRESENT

56 57 58 59 60 61

Submit

Fig.7.17: Attendance page for theory subject

Microprocessor Lab

Enter Today's Lecture Topic

Select Total Hours

Select Today's Lecture Date

Select Time Slot

Select Group

Select a Mode

56

57

58

59

60

61

Fig.7.18: Attendance page for lab subject

Hello,
test_teacher

SELECT A SUBJECT TO VIEW LECTURE DETAILS.

S.No	Lecture Topic	Lecture Hour	Lecture Start time	Lecture End time	Lecture Date	Total Student	Present	Absent	Action
1	Introduction to Computer Network	1	09:35	10:35	2022-12-08	6	4	2	<input type="button" value="Delete"/>
2	Application of CN	2	09:35	11:35	2022-12-07	6	5	1	<input type="button" value="Delete"/>
3	Limitations of CN	1	09:50	10:50	2022-12-06	6	6	0	<input type="button" value="Delete"/>

Fig.7.19: Lecture detail page

CHAPTER 8

CONCLUSION

Department Management System is a web application that helps in automating the existing manual system for maintaining records mainly related to student's attendance, syllabus, teacher's lecture plans, etc in our department. It can be monitored and controlled remotely at any time. All the users can get the required information without delay. It reduces the manpower required. It provides accurate and error free information that is not always possible with manual system. Malpractice can be reduced. All years together gathered information can be saved and can be accessed at any time. The data which is stored in the database about the students and lecture plans may be used in taking decisions by the co-ordinator.

This system is developed in understandable approach which can be understood and operated easily by all the users. This system is developed totally GUI based. This web-based application can also be used by other university departments having similar needs after applying minor changes.

8.1 Future Scope

We are also planning to add more features to our application in future to make it more feasible. It is an open source project whose source code is available on GitHub and can be edited and transformed according to the future needs.

CHAPTER 9

REFERENCES

- [1] A.Jhanvi, S.Renuka and et.al, "DEPARTMENT MANAGEMENT SYSTEM - WEB-BASED APPLICATION," *Int.Journal of Eng.Applied Sciences and Technology*, vol. 5, no. 2, pp. 356-361, 2020.
- [2] D.Kartiki, H.Neha, F.Nidhi and et.al, "Online College Management System," *Int.Journal of Computer Science and Mobile Computing*, vol. 5, no. 4, pp. 118-122, 2016.
- [3] A.V.Sinhasane, A.N.Kashid and et.al, "College Department Managemant System," *Int.Reasearch Journal of Eng. and Technology(IRJET)*, vol. 5, no. 2, pp. 214-216, 2018.
- [4] P.Krithi and Dr.M.Ramakrishna, "Student Management System - A Survey," *Int.Research Journal of Computer Science(IRJCS)*, vol. 4, no. 5, pp. 211-215, 2017.