CHAPTER 1

INTRODUCTION

1.1 Overview

Massive Open Online Courses (MOOCs) are online courses available for anyone to enroll. MOOCs provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences. Some of MOOC Course websites which offers online certification courses are:

- Swayam- https://swayam.gov.in/
- NPTEL- https://onlinecourses.nptel.ac.in/
- Mooc- http://mooc.org/
- Edx - https://www.edx.org/
- Coursera- https://www.coursera.org/
- Udemy - https://www.udemy.com/
- Khanacademy - https://www.khanacademy.org/
- Skillsahre - https://www.skillshare.com/
- Harvard University - https://online-learning.harvard.edu/
- Ted - https://ed.ted.com/
- Alison - https://alison.com/
- Futurelearn - https://www.futurelearn.com/

This mini project Digital Achievement Repository **enables the users both Student and Faculty to upload details of the MOOC course including course completed certificates. Thereby this mini project will be very useful for the department to organize and manage the MOOC details for various committees purpose**. These operations are providing with insert, delete, update and view their course details with their respective login credentials which are generated at the time of registration. At the admin side admin will have an option of managing and generating report of the same.

This application also includes the following features:

- File uploading

- Searching
- Exporting the data into csv files
- Graph generation

1.2 Objectives

The main objective of our project are as follows:

- To create a Database application for managing the MOOC course completed by Students and Staff.
- To organize the course completed by student and faculty with completed details including the Course completion certificated.
- To generate the report of the completed course by students and faculty with various filters and graph.
- To collect and store the user logs in the Database for any verification.

1.3 Organization of Project Report

The report is organized in following manner: -

- The ***chapter 1*** gives the introduction of this project that is what the project does and how it is helpful.
- The ***chapter 2*** gives the literature review of this project.
- The ***chapter 3*** gives the requirements specification of the project and this chapter briefs out the requirements required to fulfill the project.
- Th***e chapter 4*** gives the design and implementation of the project, with the help of the ER and schema diagrams, project flow can be understood clearly and this chapter also includes pseudocode.
- The ***chapter 5*** gives the snapshots of the project.
- The ***Result and Conclusion*** gives the results of the project with brief description and overall project performance analysis.

****

CHAPTER 2

LITERATURE REVIEW

The increasing popularity of **Massive Open Online Courses (MOOCs)** has significantly transformed modern education by providing flexible, low-cost, and widely accessible learning opportunities to global learners. These platforms offer certifications upon course completion, allowing students and faculty to enhance their academic and professional profiles. However, with the growing number of online certifications, educational institutions face substantial challenges in **organizing, verifying, and maintaining** these achievements in a structured manner.

Traditional digital learning platforms such as **Learning Management Systems (LMS)** primarily focus on course delivery, assessments, and classroom management. While LMS tools aid in tracking internal academic progress, they are **not designed to manage external achievements**, such as certificates earned through platforms like Coursera, NPTEL, SWAYAM, or Udemy. As a result, students and faculty often store their certificates in local devices, email attachments, or scattered folders, leading to poor accessibility and difficulty during audits or accreditation processes.

Recent developments in the field of digital credential management highlight the need for **centralized repositories** for academic achievements. Some institutions have attempted to use cloud storage systems or manual database tools to maintain certificate records, but these solutions lack **role-based access**, **searchable interfaces**, **file validation**, **analytics**, and **institution-level reporting**. Research also emphasizes the importance of integrating **analytics and visualization tools** to observe learning trends, growth patterns, and user participation statistics across semesters and departments.

Despite advancements in digital education tools, the literature shows a gap in systems specifically focused on **achievement repositories** that store, update, and analyze certificates from multiple digital learning sources. Existing platforms either:

- ➢ do not support user-driven certificate uploads,
- ➢ lack centralized access for administrators, or

- provide no insights into institutional learning trends.

The **Digital Achievement Repository** addresses these limitations by providing a **dedicated, web-based system** for storing and managing online learning achievements of both students and faculty.

It integrates features such as certificate uploads, role-based dashboards, search and filtering options, graphical analytics, and exportable reports—capabilities that are not fully present in conventional LMS systems or basic digital storage tools.

This platform bridges the gap identified in existing literature by offering a **structured, centralized, and analytics-supported achievement management system** tailored for educational institutions. It modernizes academic documentation, improves accessibility, and supports accreditation processes through organized and data-driven record management.

****

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 Functional and Non-Functional Requirements

Functional Requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior and outputs. Functional Requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what the system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements, which impose constraints on the design or implementation such as performance requirements, security or reliability.

Non-Functional Requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be Non-functional requirements are often called qualities of a system.

- **Security**: the program should not harm user’s data, corrupt files or cause any viral activity in the event of program crash. The application should not allow users to access the codes of other users for modification.
- **Ease of Use**: A reasonably good interface has to be provided. The user should be comfortable in using the application. The application should not confuse the user with too many options in a single screen.
- **Efficiency**: The software must maintain the resources judiciously the response time must be less than 5 seconds.
- **Reliability**: The system should be reliable and must not degrade the performance of the existing system and should not lead to hanging of the system.

3.2 System Requirements

3.2.1 Hardware Requirements

Computing	:	Minimum Dual Core Processor
Storage	:	Minimum 1.5GB Hard Disk space.
Memory	:	Minimum 1GB RAM.

3.2.2 Software Requirements

Server Side:

Operating system	:	Windows 8+, Ubuntu 14.04+
Server	:	WAMP / LAMP / XAMP
Front end	:	HTML, CSS, JS
Back end	:	MySQL and PHP
Text Editor	:	Notepad++ / Any Editors
Browser	:	Chrome / Firefox

Client Side:

Device	:	Smart Phone / Laptop / Desktop
Browser	:	Chrome / Firefox

****

CHAPTER 4

DESIGN AND IMPLEMENTATION

This chapter explains the detailed design and implementation of the Digital Achievement Repository. It describes the overall system architecture, database design, module-wise implementation, data flow, and backend logic used in developing the application. The chapter also highlights how different components interact to provide a secure and efficient achievement management system. Proper design ensures scalability, maintainability, and ease of use.

4.1 System Design Overview

System design represents the structural planning of the Digital Achievement Repository before implementation. It focuses on defining how different system components interact with each other to fulfill functional requirements. The design ensures that users can easily upload, manage, and retrieve achievement details through a well-organized interface. A properly planned design also reduces system complexity and improves performance.

The system adopts a **three-role model** consisting of Students, Faculty, and Admin. Each role is provided with specific access privileges and dashboards. This role-based design ensures data privacy and prevents unauthorized access. The design also supports future enhancements without affecting existing functionalities.

4.1.1 Overall Architecture

The overall architecture of the Digital Achievement Repository follows a **three-tier architecture model**. This approach separates the system into independent layers, improving modularity and maintainability. Each layer performs a specific function and communicates with other layers through well-defined interfaces.

The three-tier architecture reduces dependency between components, making the system easy to upgrade and debug. It also enhances security by restricting direct access to the database layer.

\

4.1.2 Presentation Layer

The presentation layer represents the user interface of the system. It includes web pages developed using **HTML and CSS**, allowing users to interact with the application through forms and dashboards. This layer focuses on providing a clean, user-friendly, and responsive interface.

All user inputs, such as login credentials and certificate details, are collected through this layer. It ensures proper display of data retrieved from the backend and improves the overall user experience.

4.1.3 Application Layer

The application layer is the core processing unit of the system and is implemented using **PHP**. This layer handles authentication, session management, form validation, and business logic. It processes user requests and communicates with the database to fetch or update data.

The application layer ensures that only valid data is processed and stored. It also enforces role-based access control, redirecting users to the appropriate dashboards based on their roles.

4.1.4 Data Layer

The data layer is responsible for storing and managing all system data. It is implemented using a **MySQL relational database**, which ensures structured data storage and quick retrieval. This layer stores login credentials, course details, certificate file paths, and login activity records.The relational structure of the database minimizes redundancy and ensures data integrity. It also supports complex queries required for searching, filtering, analytics, and report generation.

4.2 ER Diagram

The Entity Relationship (ER) diagram represents the logical structure of the database and the relationships among different entities. It provides a conceptual view of how data is connected and organized. The ER diagram simplifies database design by clearly identifying entities and their attributes. In the Digital Achievement Repository, the ER diagram defines entities such as students, faculty, courses, and login logs. These entities are interconnected to ensure accurate mapping of achievements to users.

.

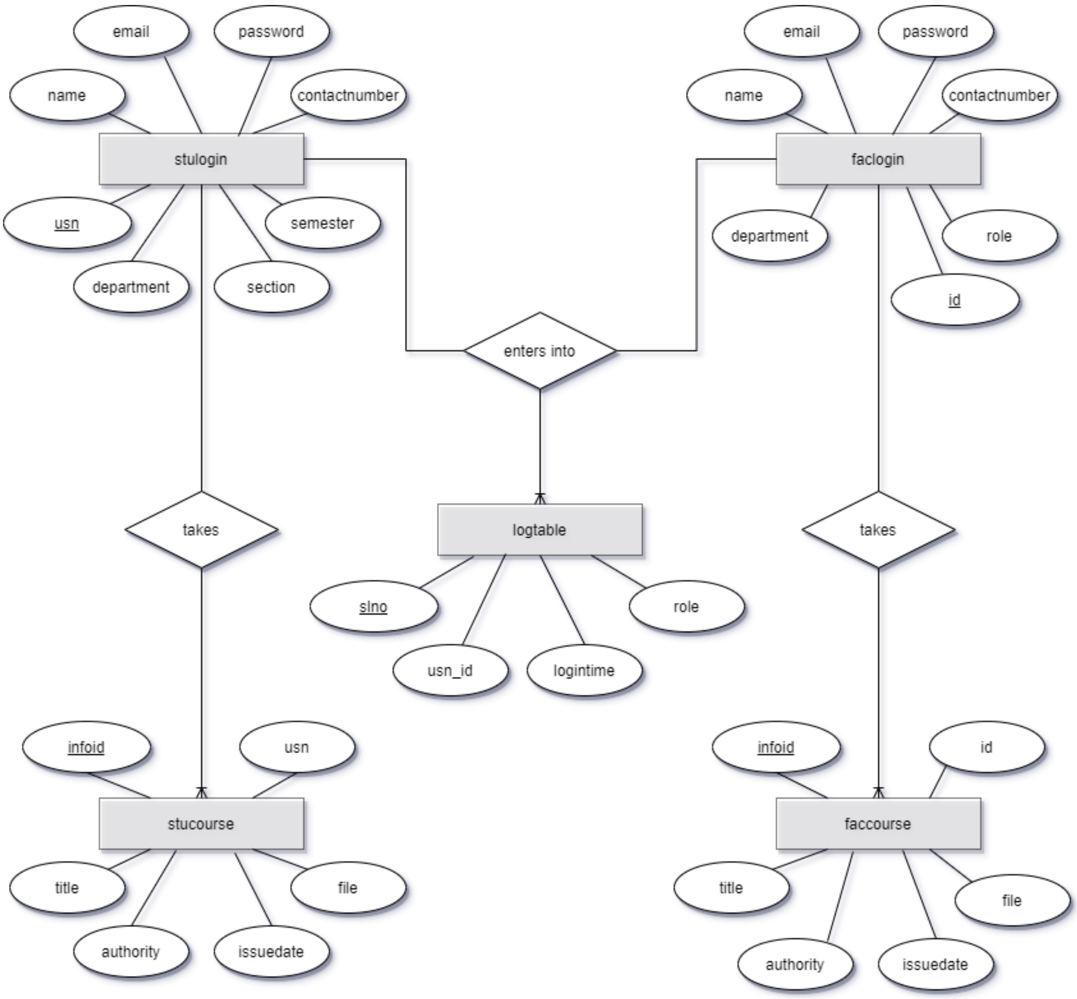
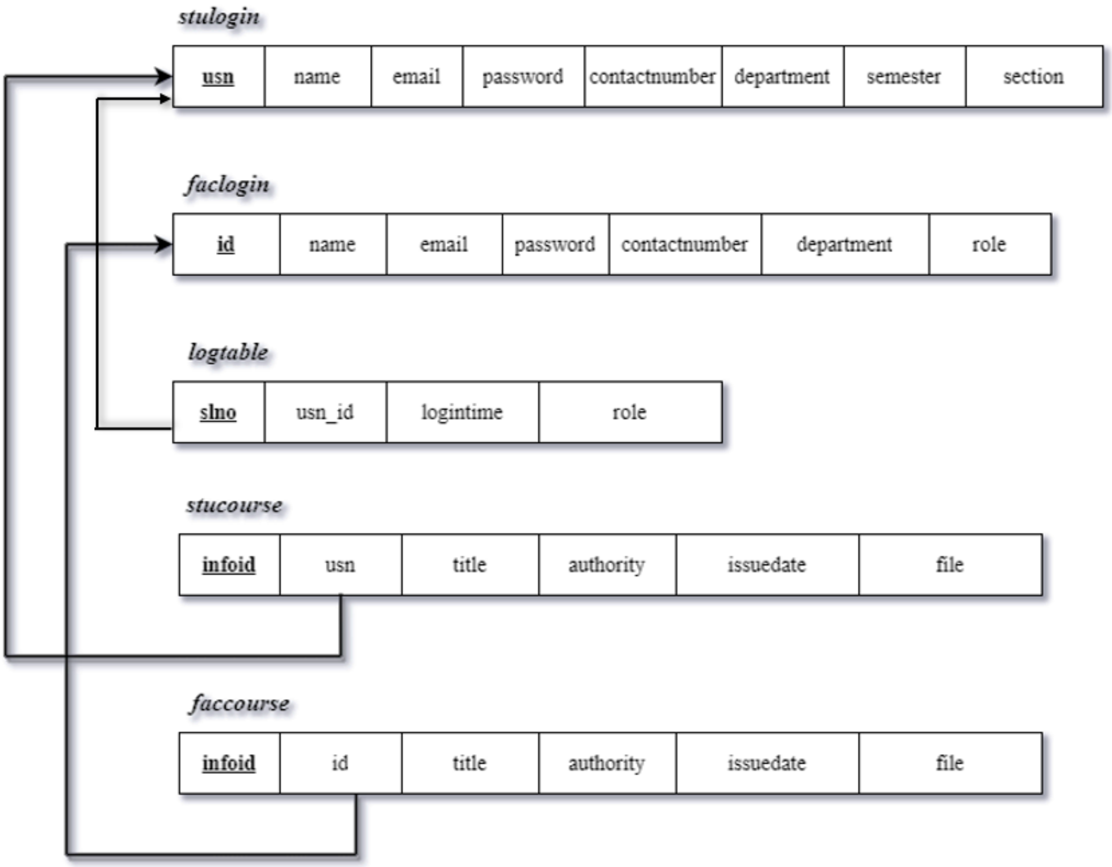


Fig 4.1: ER diagram

4.3 Schema Diagram

The database schema defines the structure of tables, columns, data types, and relationships. It translates the ER diagram into actual database tables. Proper schema design ensures efficient data storage and retrieval.

The Digital Achievement Repository database contains multiple normalized tables to reduce redundancy. Each table is designed for a specific purpose to maintain clarity and integrity.



stulogin

usn	name	email	password	contactnumber	department	semester	section

faclogin

id	name	email	password	contactnumber	department	role

logtable

slno	usn_id	logintime	role

stucourse

infoid	usn	title	authority	issuedate	file

faccourse

infoid	id	title	authority	issuedate	file

Fig 4.2:Schema diagram

4.4 Data Flow Description

Data flow describes how information moves between users, system components, and the database. It ensures smooth communication between modules. The Digital Achievement Repository follows a structured data flow starting from user input to database processing and output display.

4.4.1 Login Data Flow

When a user submits login credentials, the system validates them against the database. If credentials match, a session is created and the user is redirected to the respective dashboard.

Invalid credentials result in an error message without creating a session.

4.4.2 Certificate Upload Data Flow

Users submit certificate details through a form along with a PDF file. The application layer validates the file and moves it to the designated directory.

The certificate metadata and file path are then stored in the database.

4.4.3 Search and Filter Data Flow

The admin or user enters search criteria, which is converted into SQL queries. The database returns filtered results to the application layer.

These results are then displayed dynamically on the dashboard.

4.5 Module Design

The system is divided into independent modules to improve clarity and reusability. Each module handles a specific functionality within the application.

Modular design simplifies debugging and future enhancements.

4.5.1 Authentication Module

This module manages student, faculty, and admin login functionality. It validates credentials and creates secure sessions.

It ensures that only authorized users can access protected pages.

4.5.2 Certificate Management Module

This module handles uploading, viewing, updating, and deleting achievement records. It supports secure PDF file handling.

Users can manage their achievements independently through this module.

4.5.3 Search and Filter Module

This module enables searching course records using multiple criteria. It improves data retrieval efficiency. Admins use this module extensively for reporting and audits.

4.5.4 Analytics Module

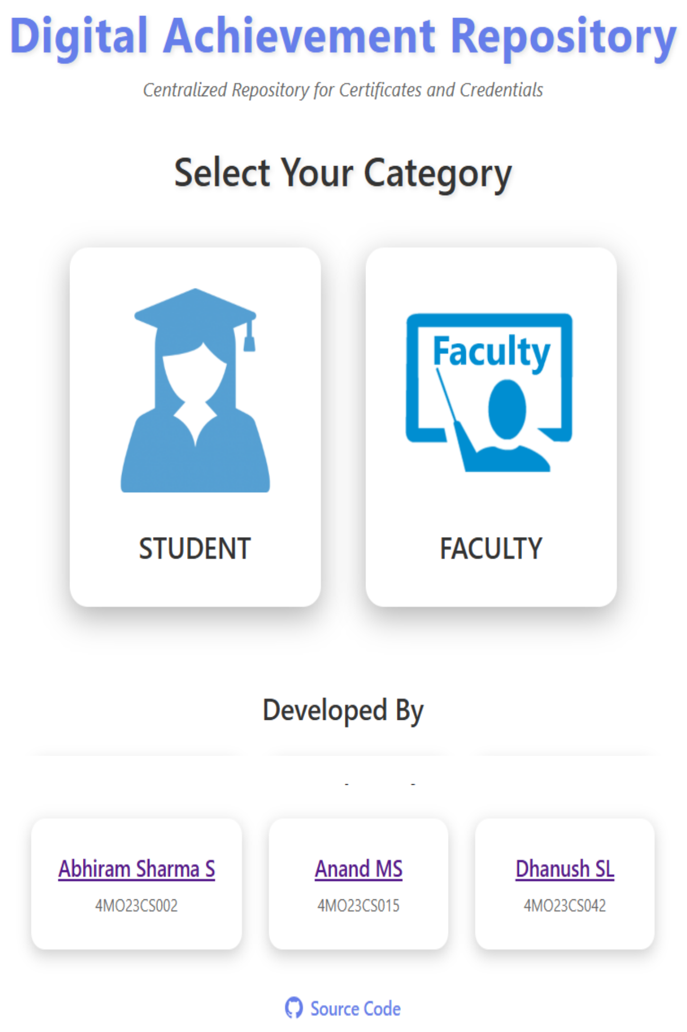
This module generates graphical statistics using Google Charts. It provides insights into learning trends.

Data visualization helps administrators make informed decisions.

****

CHAPTER 5

SNAPSHOTS



Digital Achievement Repository

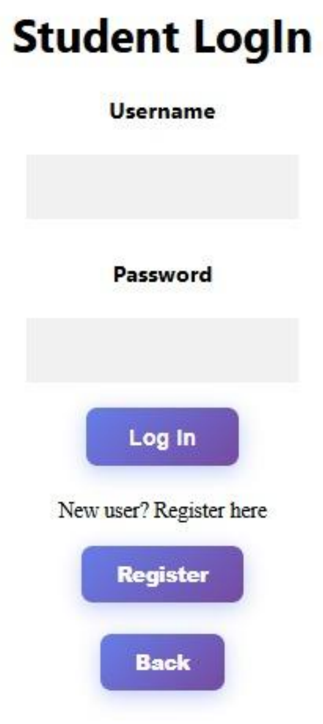
Centralized Repository for Certificates and Credentials

Select Your Category

Developed By

Abhiram Sharma S	4MO23CS002
Anand MS	4MO23CS015
Dhanush SL	4MO23CS042

Fig 5.1: Index Page



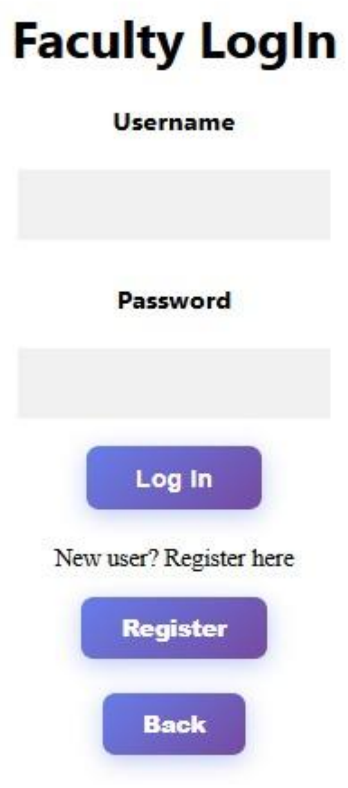
Student LogIn

Username

Password

New user? Register here

Fig 5.2(a): Student Login Form



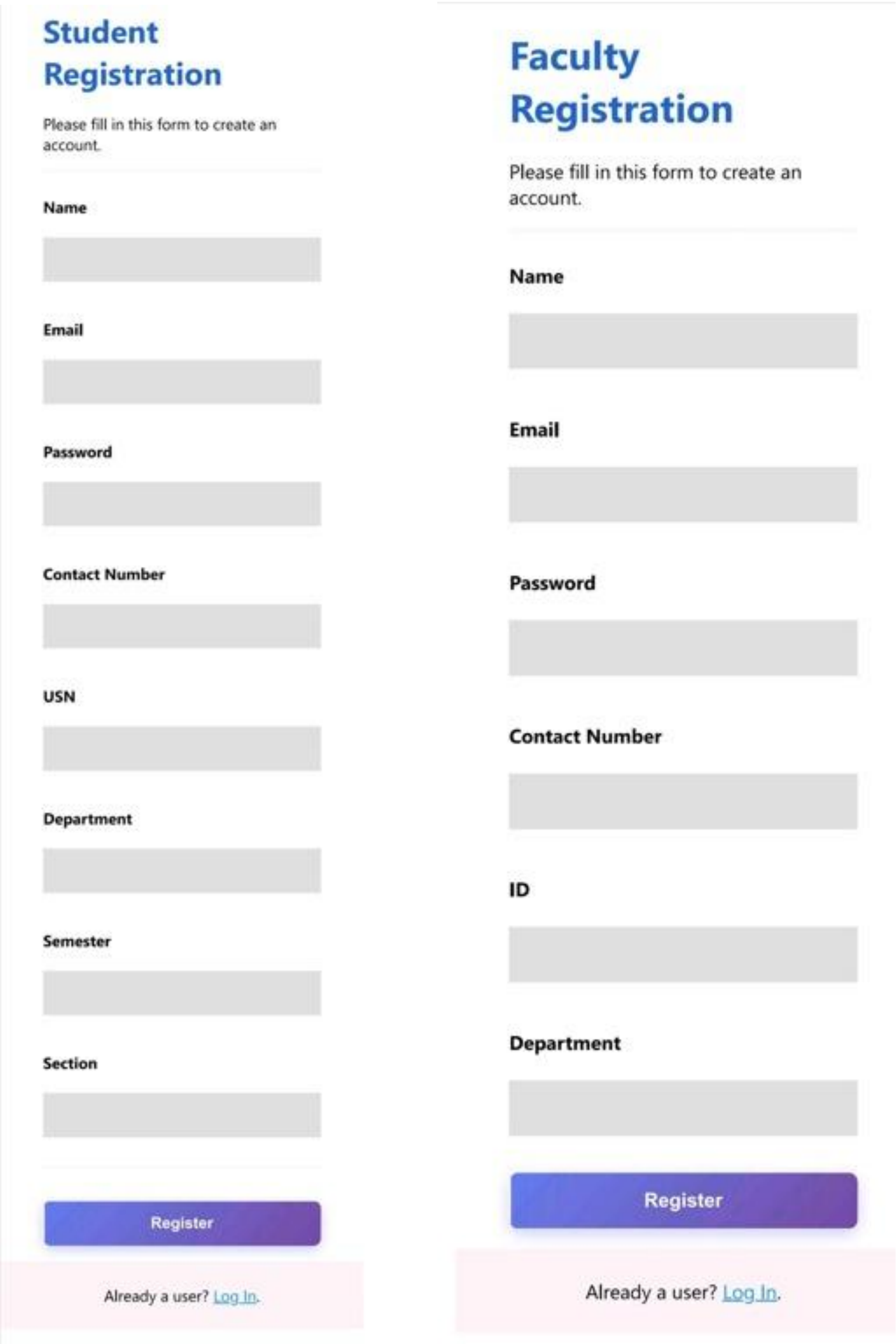
Faculty LogIn

Username

Password

New user? Register here

Fig 5.2 (b) : Faculty Login Form



Student Registration

Please fill in this form to create an account.

Name

Email

Password

Contact Number

USN

Department

Semester

Section

Register

Already a user? Log In.

Faculty Registration

Please fill in this form to create an account.

Name

Email

Password

Contact Number

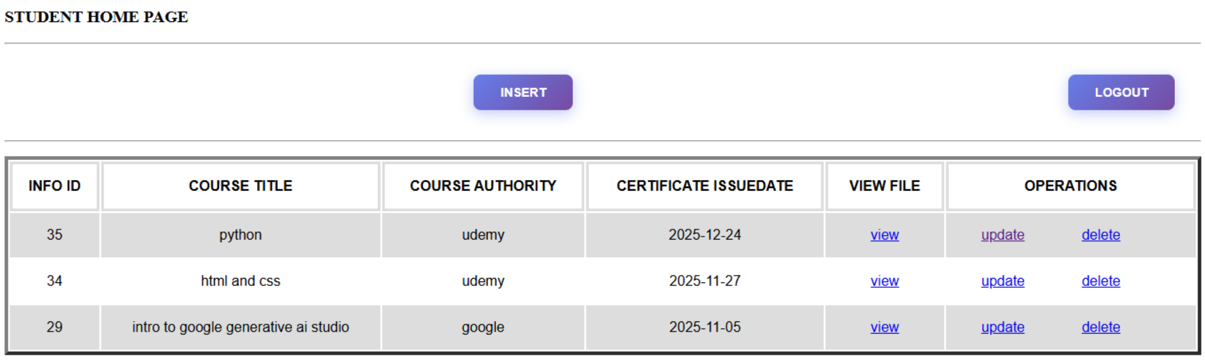
ID

Department

Register

Already a user? Log In.

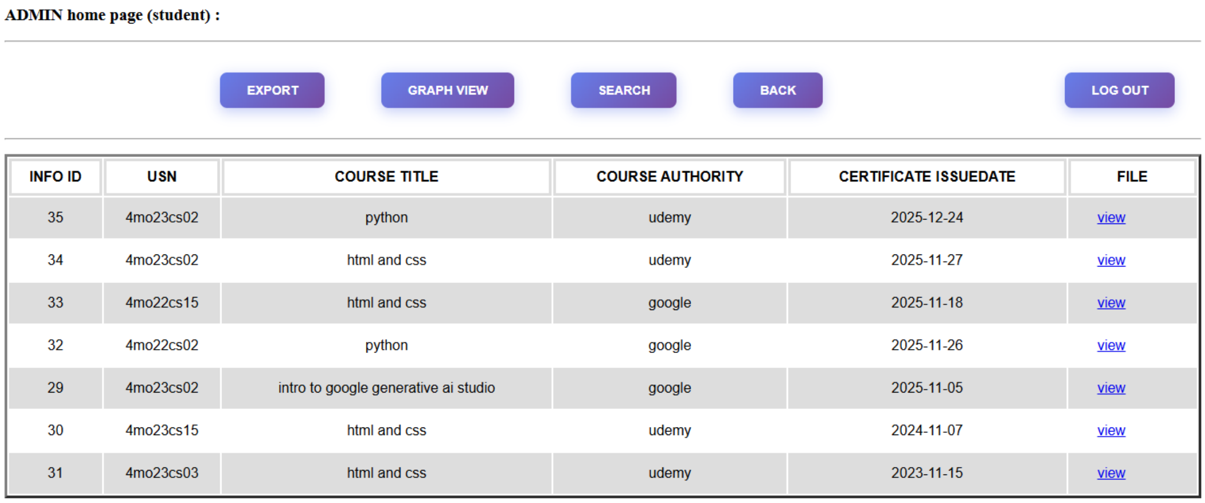
Fig 5.3: Student and Faculty Registration Form



STUDENT HOME PAGE

INFO ID	COURSE TITLE	COURSE AUTHORITY	CERTIFICATE ISSUEDATE	VIEW FILE	OPERATIONS	
35	python	udemy	2025-12-24	view	update	delete
34	html and css	udemy	2025-11-27	view	update	delete
29	intro to google generative ai studio	google	2025-11-05	view	update	delete

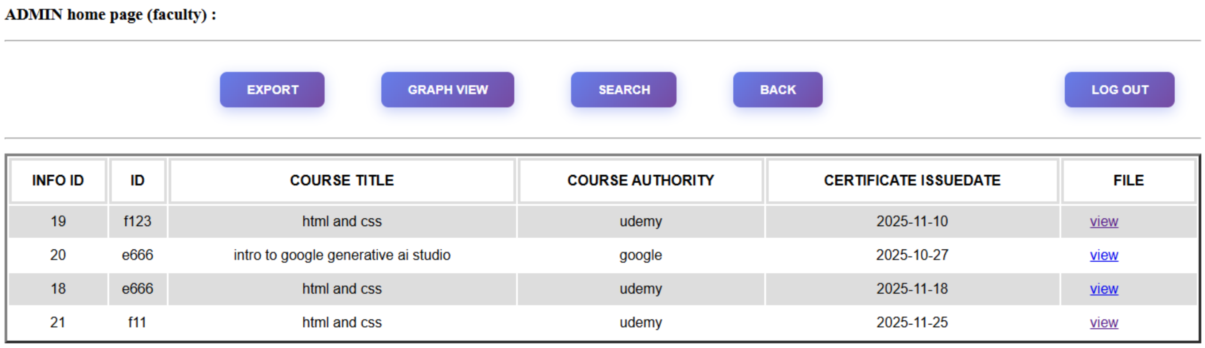
Fig 5.4: User Home Page



ADMIN home page (student) :

INFO ID	USN	COURSE TITLE	COURSE AUTHORITY	CERTIFICATE ISSUEDATE	FILE
35	4mo23cs02	python	udemy	2025-12-24	view
34	4mo23cs02	html and css	udemy	2025-11-27	view
33	4mo22cs15	html and css	google	2025-11-18	view
32	4mo22cs02	python	google	2025-11-26	view
29	4mo23cs02	intro to google generative ai studio	google	2025-11-05	view
30	4mo23cs15	html and css	udemy	2024-11-07	view
31	4mo23cs03	html and css	udemy	2023-11-15	view

Fig 5.5 (a): Admin Home Page (Student)



ADMIN home page (faculty) :

INFO ID	ID	COURSE TITLE	COURSE AUTHORITY	CERTIFICATE ISSUEDATE	FILE
19	f123	html and css	udemy	2025-11-10	view
20	e666	intro to google generative ai studio	google	2025-10-27	view
18	e666	html and css	udemy	2025-11-18	view
21	f11	html and css	udemy	2025-11-25	view

Fig 5.5 (b): Admin Home Page (Faculty)

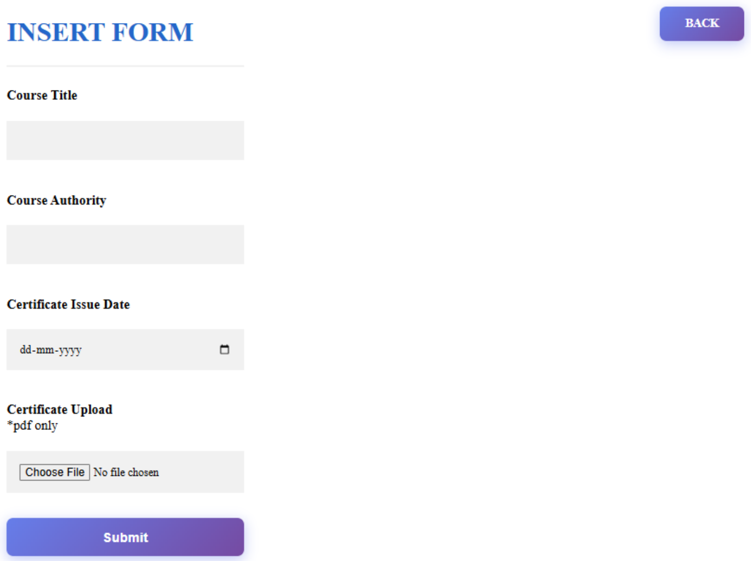


Fig 5.6: Insert Form

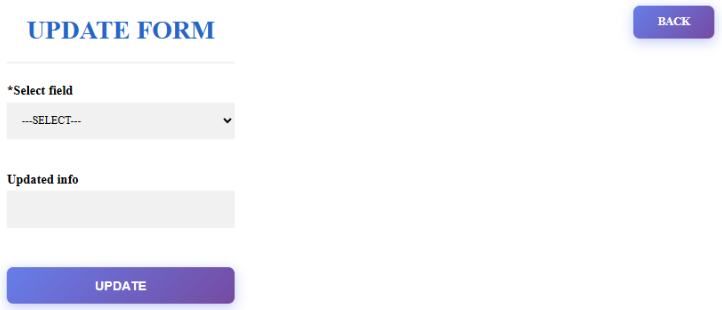
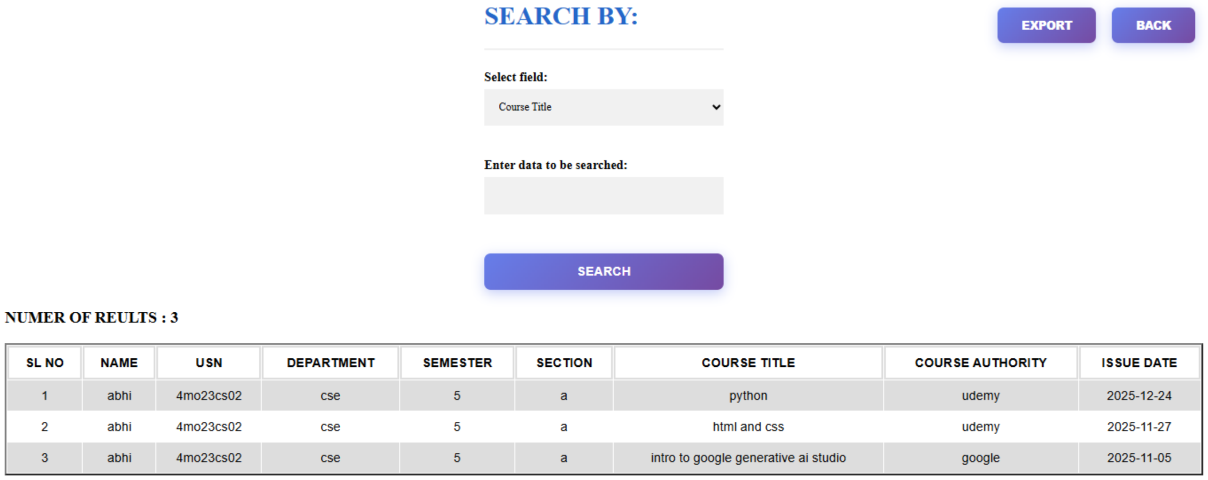


Fig 5.7: Update Form



NUMER OF REULTS : 3

SL NO	NAME	USN	DEPARTMENT	SEMESTER	SECTION	COURSE TITLE	COURSE AUTHORITY	ISSUE DATE
1	abhi	4mo23cs02	cse	5	a	python	udemy	2025-12-24
2	abhi	4mo23cs02	cse	5	a	html and css	udemy	2025-11-27
3	abhi	4mo23cs02	cse	5	a	intro to google generative ai studio	google	2025-11-05

Fig 5.8: Search Form

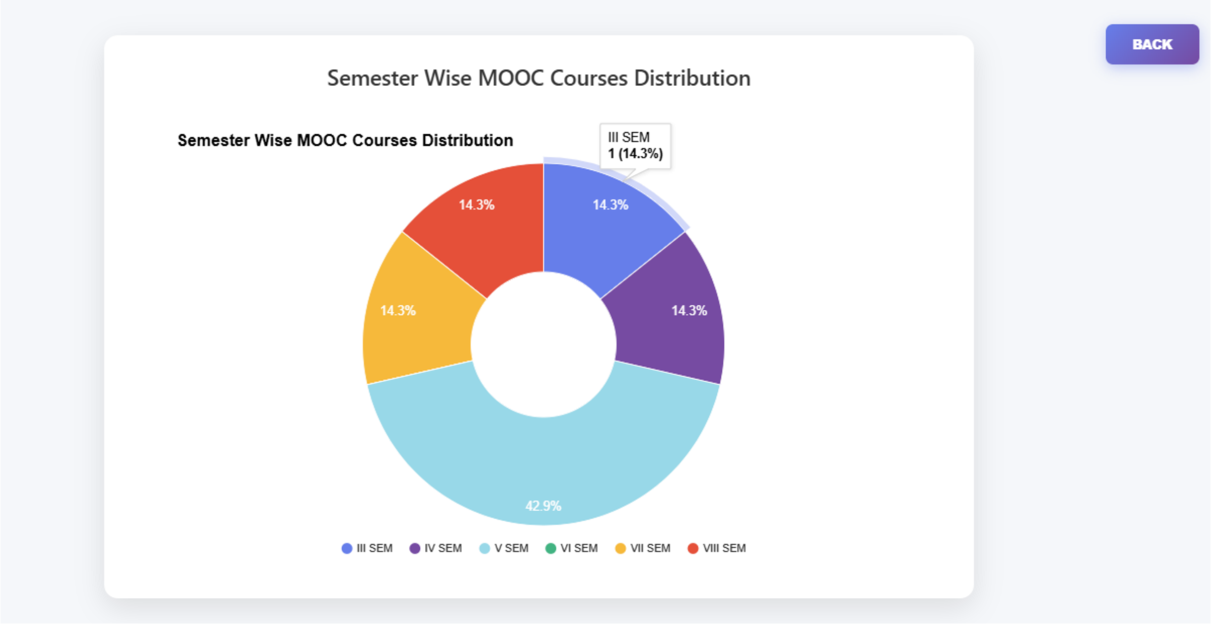


Fig 5.9 (a): Student Semester Wise Statistics

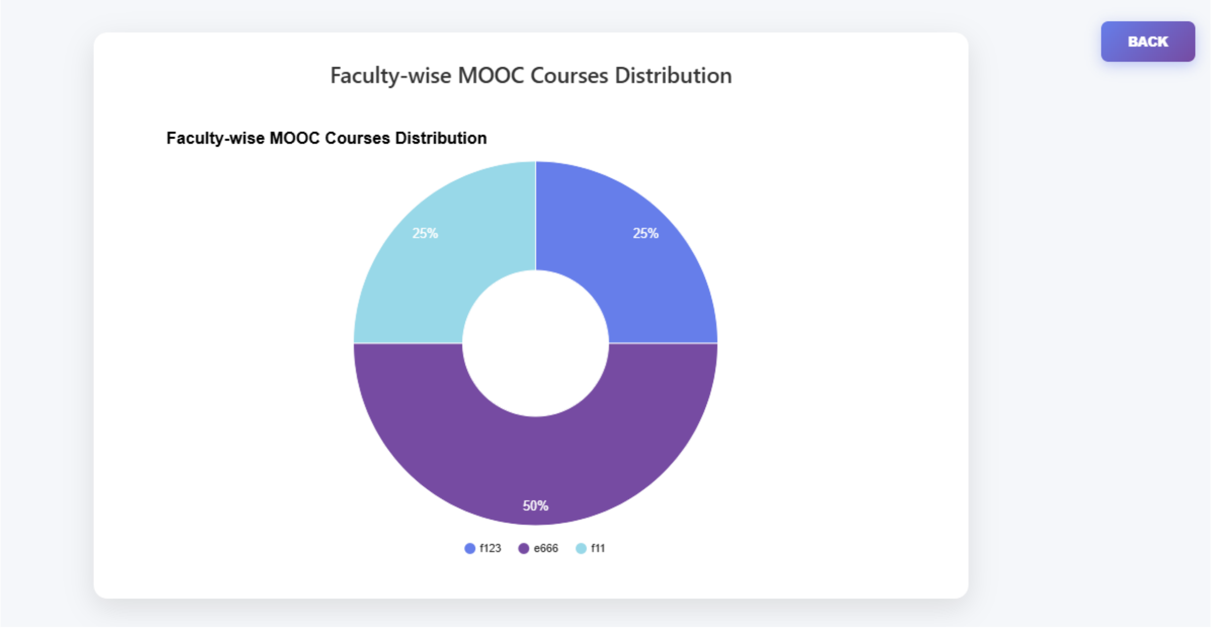

BACK
Faculty-wise MOOC Courses Distribution
Faculty-wise MOOC Courses Distribution
25%
25%
50%
f123
e666
f11

Fig 5.9 (b): Faculty Wise Statistics

****

CONCLUSION

The project titled "*Digital Achievement Repository*" is designed and developed for real time environment. It also provides a framework that enables the student and faculty to manage their data in effective way. The project has fulfilled all the objectives identified and provides an attractive user friendly interface.

Outcomes of this Project:

- **Application Perspective:**
  - User can upload and download their certificates.
  - Modifications for the details are provided to the users.
  - Admin has complete authority over the details uploaded by the users with report generation and graphical view of the data.
  - Admin can also export the details of the users to csv format.
- **Technical:**
  - Explored front end designing concepts.
  - PHP and MySQL commands.
  - Deployment.
- **Non-Technical:**
  - Requirement Analysis of Real-time Problem.
  - Report Writing.
  - About various MOOC Course.

****

REFERENCES

[1]. *Massive Open Online Courses (MOOCs) [Online], available at URL: https://www.mooc.org/*

[2]. Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*,7th Edition, 2017, Pearson, Chapter 1, page no-100-115.

[3]. File uploading [Online], available at URL: https://www.codexworld.com/upload-store-image-file-in-database-using-php-mysql

[4]. Registration from [online],available at URL: https://www.w3schools.com/howto/howto_css_register_form.asp

[5]. Graph generation [online] available at URL: https://developers.google.com/chart/interactive/docs/gallery/columnchart

[6]. PHP Official Documentation [Online], available at: https://www.php.net/docs.php

[7]. MySQL Documentation [Online], available at: https://dev.mysql.com/doc/

[8]. CRUD Operations using PHP and MySQL [Online], available at: https://www.javatpoint.com/php-crud

****