**FACULTY MANAGEMENT SYSTEM**



Mini Project submitted in partial fulfillment of the requirement for the award of the

Degree of

#### BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE AND ENGINEERING**

Under the esteemed guidance of

#### M. Vishwashanthi Assistant Professor

By

**A. Nagalaxmi 21R11A05A9**

**G. Madhukiran 21R11A05C4**

**A. Sai Bhagya Sree 21R11A05E3**



### Department of Computer Science and Engineering

**Accredited by NBA**

### Geethanjali College of Engineering and Technology

#### (UGC Autonomous)

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi) Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

**September-2024**

**Geethanjali College of Engineering & Technology**

#### (UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi) Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

#### Accredited by NBA



**CERTIFICATE**

This is to certify that the B.Tech Mini Project report entitled **“Faculty Management System”** is a bonafide work done by **A. Nagalaxmi (21R11A05A9), G. Madhukiran (21R11A05C4), A. Sai Bhagya Sree (21R11A05E3)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in ―**Computer Science and Engineering**‖ from Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

Internal Guide HOD – CSE

#### M. Vishwashanthi Dr. A. SreeLakshmi

Assistant Professor Professor

**Geethanjali College of Engineering & Technology**

#### (UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi) Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

#### Accredited by NBA



**DECLARATION BY THE CANDIDATE**

We, **A. Nagalaxmi, G. Madhukiran, A. Sai Bhagya Sree**, bearing Roll Nos. **21R11A05A9, 21R11A05C4**, **21R11A05E3** hereby declare that the project report entitled **“Faculty Management System”** is done under the guidance of **Mrs. M. Vishwashanthi, Assistant Professor,** Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

**A. Nagalaxmi (21R11A05A9) G.Madhukiran(21R11A05C4)**

**A. Sai Bhagya Sree (21R11A05E3)**

**Department of CSE, Geethanjali College of Engineering and Technology,**

**Cheeryal.**

**ACKNOWLEDGEMENT**

We, the students of the Computer Science and Engineering Department at Geethanjali College of Engineering and Technology, would like to express our sincere gratitude to the college authorities for providing the necessary resources and facilities to successfully complete our mini project, titled **“Faculty Management System.”**

We are deeply thankful to **Prof. Dr. S. Udaya Kumar**, Principal of Geethanjali College of Engineering and Technology, for his invaluable guidance and continuous encouragement throughout this project. His insightful advice and unwavering commitment to fostering a culture of innovation and learning have been instrumental in the successful completion of our work.

We also extend our heartfelt appreciation to **Dr. A. SreeLakshmi, Professor**, and Head of the Department of Computer Science and Engineering, for her dedicated support and mentorship. Her leadership and commitment to our academic and professional development have been a source of motivation and guidance throughout this project.

A special thanks to **Mrs. M. Vishwashanthi, Assistant Professor**, whose technical expertise and thoughtful guidance have been essential to navigating the complex aspects of this project. Her constructive feedback and willingness to share her knowledge have greatly contributed to the technical success of our work.

We are grateful to all our mentors and faculty members for their continuous support and encouragement.

#### A. Nagalaxmi (21R11A05A9) G.Madukiran (21R11A05C4)

**A. Sai Bhagya Sree (21R11A05E3)**

**ABSTRACT**

The Faculty Management System (FMS) is a streamlined digital solution designed to efficiently handle the registration and management of faculty information within an educational institution. This system allows administrators to register new faculty members by entering detailed personal and professional information into a centralized database. The FMS offers robust search functionality, enabling quick retrieval of specific faculty data based on various criteria such as name, department, or research interests. This ensures that administrative tasks and decision-making processes are supported by accurate and up-to- date information. Additionally, the FMS includes features that allow for the easy extraction and download of faculty data in Excel format. This capability facilitates reporting, data analysis, and sharing of information across departments. By providing a secure and user- friendly platform, the FMS enhances administrative efficiency, ensures data accuracy, and supports data-driven decision-making within the institution. The system also integrates role-based access control, ensuring that sensitive faculty data remains protected, with access granted only to authorized personnel. Furthermore, its scalable architecture allows it to accommodate the growing needs of institutions, making it adaptable to both small and large- scale educational-organizations.

.

**LIST OF FIGURES/DIAGRAMS/GRAPHS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Fig. No** | **Title of Figure** | **Page No** |
| 1 | 3.1 | Traditional FMS | 6 |
| 2 | 3.2 | Mobile apps for Faculty Management | 7 |
| 3 | 3.3 | Cyber security shields | 8 |
| 4 | 3.4 | Before & after effect of digital transformation | 9 |
| 5 | 3.5 | Ethical consideration in technology | 10 |
| 6 | 4.1. 1 | Three-Tier Architecture in Application | 11 |
| 7 | 4.2.1.1 | Use Case Diagram | 14 |
| 8 | 4.2.2.1 | Class Diagram | 15 |
| 9 | 4.2.3.1 | Sequence Diagram | 17 |
| 10 | 7.1 | Main Interface of the FMS | 32 |
| 11 | 7.2 | Admin Login Screen | 32 |
| 12 | 7.3 | Admin Dashboard | 33 |
| 13 | 7.4 | Add a new member Screen | 33 |
| 14 | 7.5 | Save new member | 34 |
| 15 | 7.6 | Faculty Login | 34 |
| 16 | 7.7 | Faculty Dashboard | 35 |
| 17 | 7.8 | Faculty Profile Page | 35 |
| 18 | 7.9 | Faculty Profile Update Page | 36 |
| 19 | 7.10 | Post Update Page | 36 |

|  |  |  |  |
| --- | --- | --- | --- |
| 20 | 7.11 | Hods Dashboard | 37 |
| 21 | 7.12 | Manage Faculty Page | 37 |
| 22 | 7.13 | View Faculty Page | 38 |
| 23 | 7.14 | Filter Faculty Page | 38 |
| 24 | 7.15 | Filtering Faculty Page | 39 |
| 25 | 7.16 | Filtered Faculty Page | 39 |
| 26 | 7.17 | Download Excel Sheet Page | 40 |
| 27 | 7.18 | Downloaded Excel Sheet | 40 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Abbreviation** | **Full Form** |
| 1 | FMS | Faculty Management System |
| 2 | IT | Information Technology |
| 3 | XAMPP | Cross-platform, Apache, MariaDB, PHP |
| 4 | API | Application Programming Interface |
| 5 | UI | User Interface |
| 6 | HTTP | Hypertext Transfer Protocol |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Contents** | **Page. No** |
|  | Abstract | v |
|  | List of Figures/Diagrams/Graphs | vi |
|  | List of Abbreviations | viii |
| **1** | **Introduction** |  |
|  | 1.1 About the Project | 1 |
|  | 1.2 Objectives of the Project | 1 |
| **2** | **System Analysis** |  |
|  | 2.1 Existing System | 2 |
|  | 2.2 Proposed System | 2 |
|  | 2.3 Feasibility Study | 2 |
|  | 2.3.1 Details | 2 |
|  | 2.3.2 Impact on environment | 3 |
|  | 2.3.3 Safety | 3 |
|  | 2.3.4 Ethics | 3 |
|  | 2.3.5 Cost | 3 |
|  | 2.3.6 Type | 3 |
|  | 2.3.7 Standards | 4 |
|  | 2.4 Scope of Project | 4 |
|  | 2.5 System Configuration | 4 |
| **3** | **Literature Overview** | 6 |

|  |  |  |
| --- | --- | --- |
| **4** | **System Design** |  |
|  | 4.1 System Architecture | 11 |
|  | 4.1.1 Module Description | 12 |
|  | 4.2 UML Diagrams | 14 |
|  | 4.3 System Design | 19 |
|  | 4.3.1 Modular Description | 19 |
|  | 4.3.2 Database Design | 19 |
| **5** | **Implementation** |  |
|  | 5.1 Implementation | 21 |
|  | 5.2 Sample code | 24 |
| **6** | **Testing** |  |
|  | 6.1 Software Testing | 27 |
|  | 6.2 Test Cases | 29 |
| **7** | **Outputs Screens** | 32 |
| **8** | **Conclusion** |  |
|  | 8.1 Conclusion | 41 |
|  | 8.2 Further Enhancements | 42 |
| **9** | **Bibliography** |  |
|  | 9.1 Books References | 43 |
|  | * 1. Websites References   2. Technical Publication References | 43  43 |
| **10** | **Appendices** | 44 |

1. **INTRODUCTION**

## ABOUT THE PROJECT

The Faculty Management System (FMS) is designed to streamline the administration of faculty data within educational institutions. In traditional setups, managing faculty information, including personal details, academicqualifications, and professional history, is often a time-consuming and error-prone task, primarily relying on paper-based systems or decentralised databases. FMS resolves these issues by providing a centralised digital platform where all faculty data can be efficiently stored, managed, and retrieved.

This system supports educational institutions in maintaining accurate and up-to-date faculty records, enabling administrators to quickly search and retrieve data based on specific criteria such as department, research interests, or years of service. By centralising data, the FMS facilitates smoother administration, better decision-making, and easier access to essential faculty information.

The Faculty Management System (FMS) is a comprehensive digital platform developed to handle the complete life cycle of faculty data management, from registration to retirement. The system incorporates features like detailed faculty profile creation, real-time data updates, and advanced search functionalities to ensure quick retrieval of relevant information. Additionally, FMS enables the extraction of faculty data in multiple formats, including Excel, to streamline reporting and data sharingacross departments. The system is designed to adapt to institutions of varying sizes, with scalability built into its architecture to accommodate future growth.

## OBJECTIVES OF THE PROJECT

The primary objective of the FMS is to optimize the process of managing faculty information, ensuring that data is easily accessible, secure, and up to date. It aims to enhance administrative efficiency, support decision-making with accurate data, and simplify reporting tasks by offering a platform for seamless information management.

# SYSTEM ANALYSIS

## EXISTING SYSTEM

Most educational institutions still rely on manual or semi-automated processes for faculty data management, often using spreadsheets or isolated databases. These methods are prone to errors, data redundancy, and inefficiencies. The lack of integration between different departments can result in delayed access to critical faculty information, impacting administrative tasks like faculty evaluations, workload distribution, and decision-making.

## PROPOSED SYSTEM

The Faculty Management System proposes a fully digital platform that centralizes faculty data in one secure location. With features such as role-based access control, automated data validation, and real-time updates, the FMS streamlines the administrative process, reducing the time and effort required to manage faculty information. The system includes advanced search filters that allow users to retrievedata based on specific criteria and download the information in report-ready formatslike Excel.

## FEASIBILITY STUDY

* + 1. **DETAILS**

The feasibility study assesses the technical, operational, and economic factors required for implementing the FMS. Technically, the system is feasible due to the availability of web development tools and database management systems that ensure a smooth and scalable implementation. Operationally, the FMS can significantly reduce the workload of administrators and improve decision-making processes. Economically, while there are upfront development and maintenance costs, the long-term benefits of time savings and reduced administrative errors justify the investment.

## IMPACT ON ENVIRONMENT

The FMS contributes positively to the environment by minimizing the use of paper,as all faculty records are digitized. The reduction in physical paperwork also lowersthe need for printing, ultimately reducing waste and energy consumption. Additionally, the system reduces the need for physical file storage, leading to moreefficient use of space.

## SAFETY

The FMS ensures the safety of data through robust security protocols, includingencryption and role-based access controls, to prevent unauthorized access to sensitive information. Regular security updates and audits are incorporated to ensure data protection and prevent breaches.

## ETHICS

Ethical considerations in the FMS focus on safeguarding the privacy of facultydata. The system is designed with clear user consent mechanisms for data usage, and it complies with legal standards related to data protection. The system ensures that personal information is handled transparently and is only accessible by authorized personnel.

## COST

The costs associated with the FMS primarily involve the initial development, software licensing, and ongoing maintenance. Despite these upfront costs, the long-term financial benefits include reduced administrative workload, fewer errors, and streamlined reporting processes, which will ultimately lead to savingsin time and resources.

## TYPE

The FMS is a web-based application accessible via modern web browsers on both desktop and mobile devices. It serves as a tool for educational institutions to manage faculty data effectively and can also integrate with other institutional software systems, enhancing its overall functionality.

## STANDARDS

The development of the FMS adheres to Agile methodology, allowing for iterative development and continuous improvements based on user feedback. Thesystem is designed following best practices in software development and data management to ensure scalability and reliability. It also complies with industry standards for data security and privacy.

## SCOPE OF THE PROJECT

The scope of the Faculty Management System includes the complete digitization of faculty data management processes within an educational institution. It allows administrators to register new faculty members, update information in real time, andretrieve data using advanced search filters. The project also covers features such asdata export for reporting purposes and role-based access to ensure that only authorized personnel can view or edit sensitive information. The system is scalable to accommodate the growth of the institution and future enhancements, such as integrating faculty evaluation systems or automated workload distribution.

## SYSTEM CONFIGURATION

### Software Required:

#### Server/Development Machine:

* 1. **Operating System:** Windows 10/11, macOS, or Linux (Uuntu).
  2. **Text Editor/IDE:** Visual Studio Code, Sublime Text, or PHPStrom
  3. **Web Browser:** Latest version of Chrome, Firefox, or Edge for testing
  4. **XAMPP:** XAMPP v7.4.29 (or any compatible version)
     1. Apache**:** Web server for hosting the PHP scripts
     2. PHP**:** PHP v7.4 for server-side scripting

#### Client-Side (User Access):

* 1. **Operating System:** Any modern OS (Windows, macOS, Linux)
  2. **Web Browser:** Chrome, Firefox, Safari, or Edge

### Hardware Required:

#### Server/Development Machine:

1. **Processor:** Minimum Dual-Core Processor (Intel i3 or equivalent)
2. **RAM:** 4 GB (8 GB recommended for smoother performance)
3. **Storage:** 20 GB of free disk space
4. **Network:** Broadband Internet connection (for onlinedeployment)
5. **Monitor:** Minimum resolution of 1024x768
6. **Peripherals:** Keyboard, Mouse, and a stable powersupply

#### Client-Side (User Access):

1. **Processor:** Any modern processor (Intel i3 orequivalent)
2. **RAM:** 2 GB (4 GB recommended)
3. **Storage:** 2 GB of free disk space
4. **Browser:** Latest version of Chrome, Firefox, or Edge
5. **Internet:** Stable internet connection for accessing thesystem online

# LITERATURE OVERVIEW

These papers provide a comprehensive understanding of the challenges, solutions, and best practices involved in the development and implementation of digital management systems, particularly focusing on the transition from traditional, paper-based systems to fully digital platforms in educational institutions. They offer insights into both the technical and organizational challenges, as well as the potential benefits andethical considerations.

#### Paper 1: "Digital Transformation in Higher Education: The Role of Faculty Management Systems"

Problem Identified: This paper explores the key challenges universities face in adopting digital platforms to manage faculty data. It highlights issues such as resistance to change, lack of user training, and data integration complexities. The study emphasizes the importance of thorough digital transformation strategies, including infrastructure development and training programs, to ensure the success of faculty management systems.

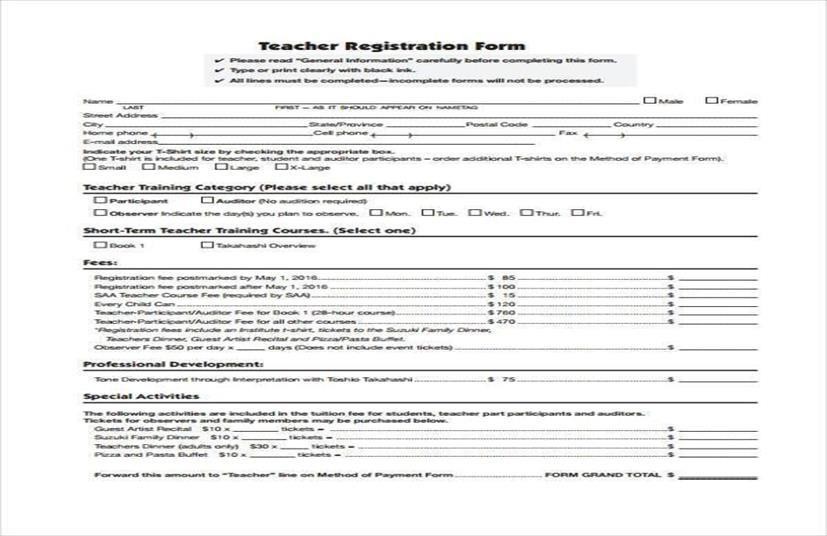


Fig 3.1 Traditional FMS

#### Paper 2: "Improving Administrative Efficiency Through Faculty Information Systems"

Problem Identified: This paper investigates the inefficiencies in traditional faculty management processes, where records are stored manually or across decentralized databases. The barriers identified include time-consuming administrative tasks, data inconsistencies, and limitedaccessibility to information. The paper argues that adopting a centralized Faculty Management System can significantly improve operational efficiency by streamlining data storage and retrieval processes.

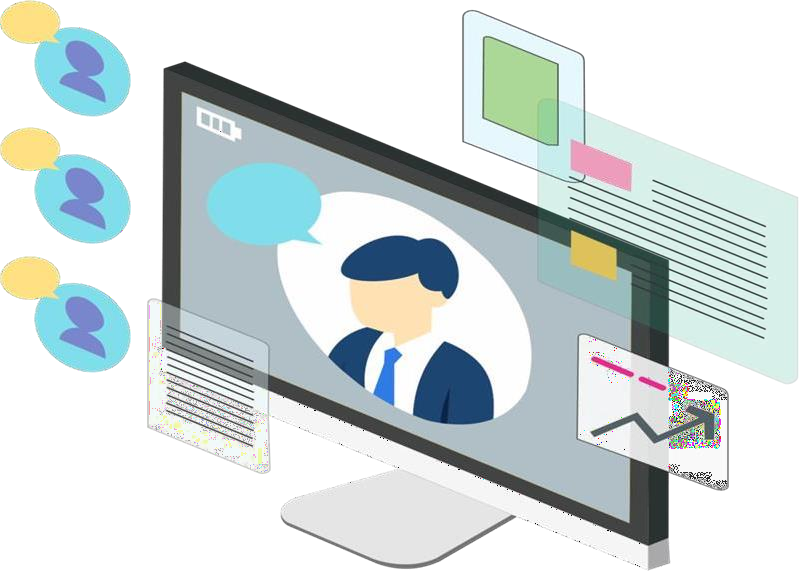


Fig 3.2 Mobile apps FMS

#### Paper 3: " Cybersecurity Challenges in Educational Systems: A Case Study ofFaculty Management Systems"

Problem Identified: This paper focuses on the cybersecurity risks associated with faculty management systems, such as unauthorized access, data breaches, and the exposure of personal information. It emphasizes the importance of implementing strong security measures, such as encryption, role-based access control, and regular security audits. The paper also outlines the consequences of cybersecurity breaches, including damage to the institution's reputation and potential legal ramifications.

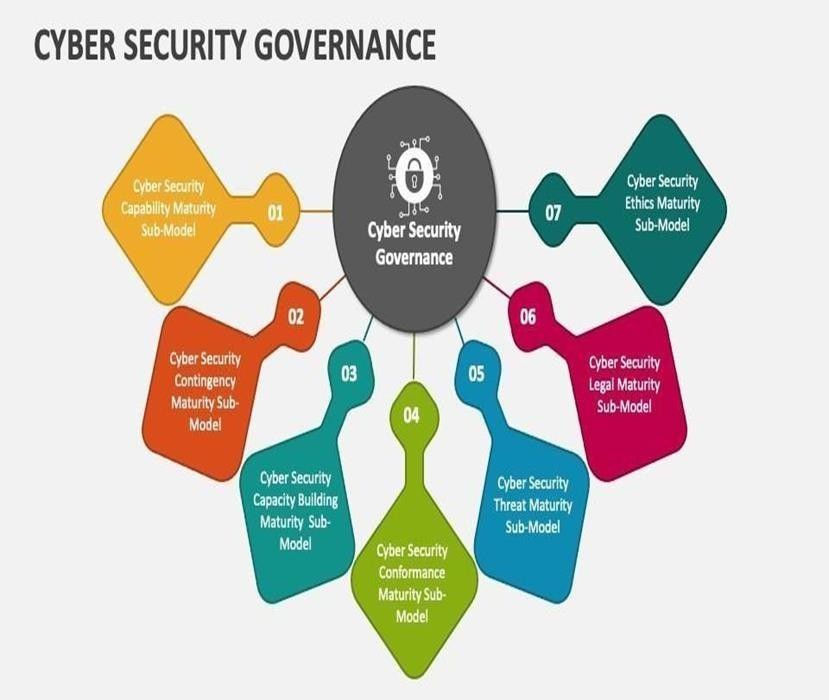


Fig 3.3 Cybersecurity shields, digital locks, or code with legal or lawenforcement symbols.

**Paper 4: " Enhancing Data Accessibility and Accuracy in Higher Education:A Comparative Study of Traditional vs. Digital Faculty Management Systems"**

Problem Identified: The paper examines the inefficiencies in traditional faculty data management systems, such as manual errors, delays in accessingrecords, and challenges in retrieving specific data. The study compares these issues with the benefits observed in institutions that have implemented digital faculty management systems, including enhanced data accuracy, quicker access to records, and streamlined administrative processes. It also explores the cost-effectiveness of such systems in the long term.

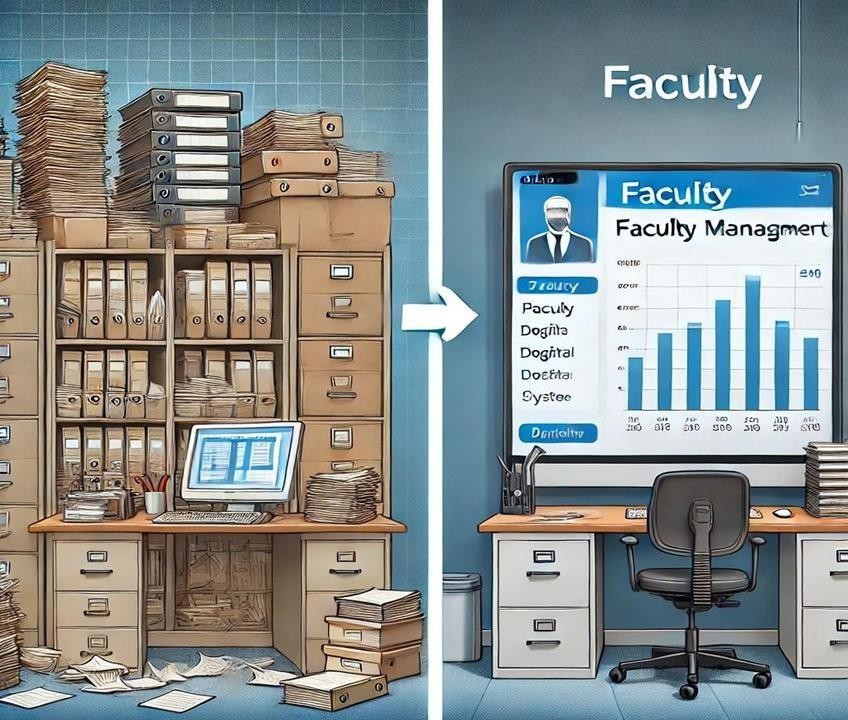


Fig 3.4 Image showing the before-and-after effect of digital transformatio

#### Paper 5: " Legal and Ethical Considerations in Use of Faculty ManagementSystems"

Problem Identified: This paper discusses the legal and ethical challengesthat arise from managing sensitive faculty data digitally, including compliance with data protection laws, maintaining confidentiality, and ensuring that data is used appropriately. It stresses the need for clear policiesregarding data access, storage, and usage, as well as ethical guidelines to protect faculty members' personal information.

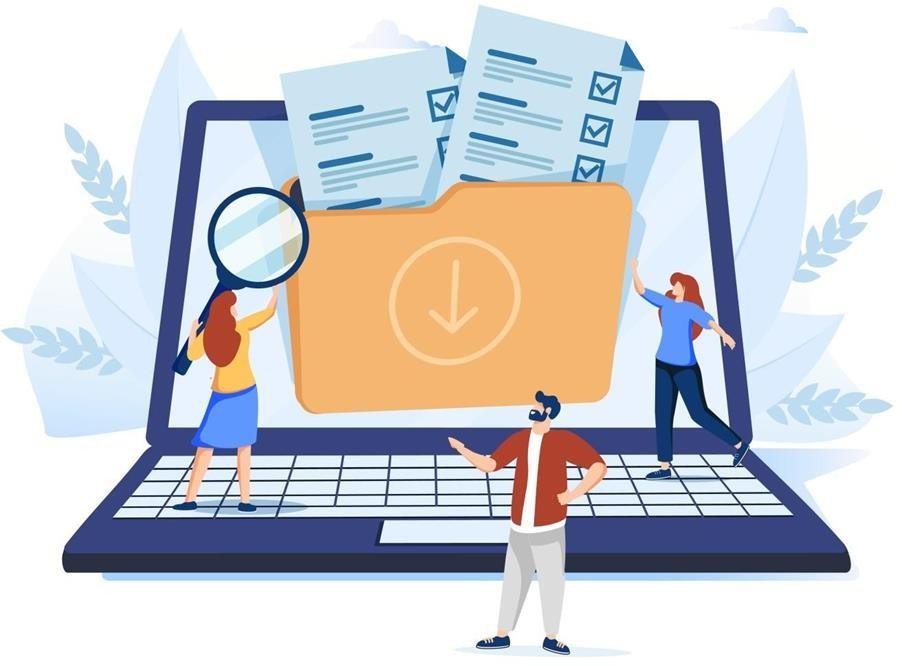
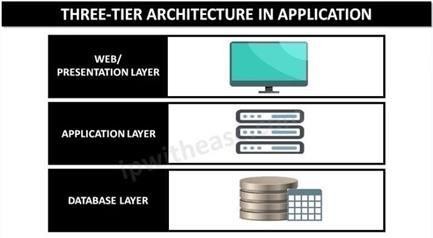


Fig 3.5 Image representing ethical considerations in FMS

# SYSTEM DESIGN

## SYSTEM ARCHITECTURE

The system architecture of the Faculty Management System (FMS) is designed to provide an efficient, secure, and scalable platform for managingfaculty data, student-faculty interactions, and administrative tasks. It follows a multi-tier architecture that ensures separation of concerns, allowing easy updates and maintenance:



#### Fig 4.1.1 Architecture of Website

1. **Presentation Layer (Frontend):**

The presentation layer is responsible for the user interface, where users such as Admin, Faculty, and HODs interact with the system. It includes login pages, profile forms, dashboards, and filtering options for HODs to search for faculty. The design is implemented using HTML, CSS, and JavaScript to ensure a clean, user-friendly experience. Each user role has customized access to different features, such as profile management forfaculty and filtering capabilities for HODs. The layout and design are optimized for seamless navigation and interaction

#### Business Logic Layer (Backend):

The business logic layer is the core of the system, handling requests from the user interface and processing them. This layer includes login authentication, session management, profile updates, and faculty filtering functionality. PHP scripts act as the

intermediary between the frontend and the database, executing actions like validating login credentials, updating user profiles, and processing filtering options based on department and criteria like age, qualification, and experience. The layer ensures that only valid requests from authenticated users are processed and stored.

#### Data Layer (Database):

The data layer stores all system information, including faculty profiles, HOD details, and admin credentials, using MySQL with multiple tables for faculty data like name, age, and qualifications. It ensures data integrity and proper table relationships. When a user updates a profile or anHOD filters faculty, this layer retrieves or stores the data.

## MODULE DESCRIPTION

### Admin Module:

The admin module manages the overall system, including user account creation, role assignments, and data oversight. Admins can add new faculty, HODs, and manage user credentials. This module includes functionalities to edit, delete, or update faculty profiles and monitor systemusage. Admins have access to view reports and handle notifications for bothfaculty and HODs. It serves as the central hub for ensuring the smooth operation of the Faculty Management System. Admins can also download faculty data for offline records or analysis.

### Faculty Module:

The faculty module allows individual faculty members to log in, update their personal profiles, and manage their teaching information. Faculty can edit details like age, specialization, experience, and upload documents such as resumes and portfolios. They also receive notifications related to their department and can view any administrative updates.The module provides the faculty with streamlined interface to track their professional information. It ensures that faculty data is always current and available for

HODs or admin review.

### HOD Module:

The HOD module enables heads of departments to manage faculty data within their specific departments. HODs can view and filter faculty profiles based on various criteria such as experience, qualifications, or specialization. They are responsible for reviewing and verifying the accuracy of the data submitted by faculty members. The module also allows HODs to communicate with faculty and send notifications regarding departmental updates. It provides essential tools for managing departmentalactivities, ensuring efficient faculty oversight.

### Authentication Module:

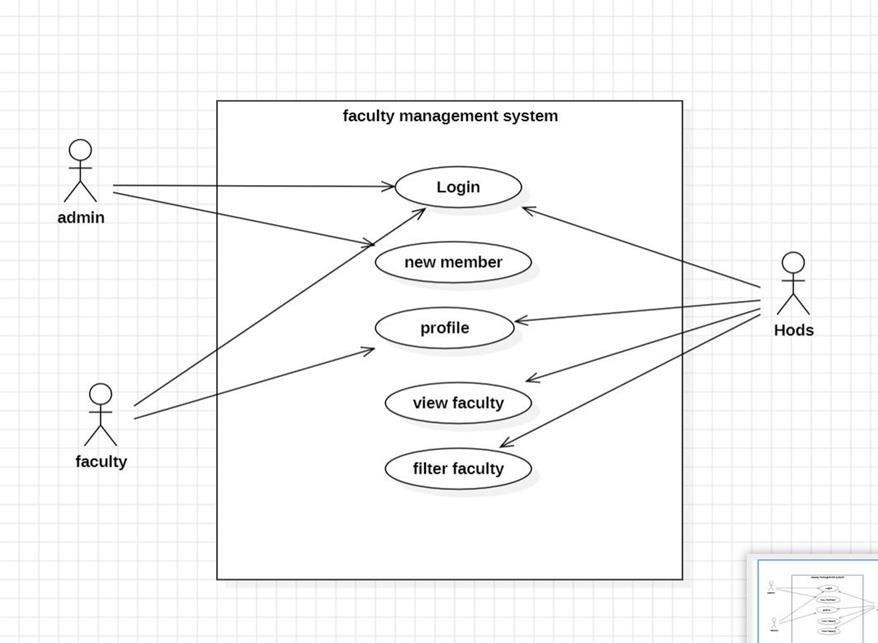
This module handles user authentication for all roles, including admin,faculty, and HODs. It verifies login credentials against stored data and grantsaccess to specific dashboards based on the user's role. The module ensures secure access to the system, protecting sensitive data like personal details and qualifications. Failed login attempts are tracked, and appropriate error messages are displayed to the user. Additionally, the module may include password recovery and user account management features. It is crucial for maintaining system security and user access control.

## UML DIAGRAMS

The Unified Modeling Language (UML) diagrams help visualize the design and structure of the faculty management system. The following diagrams are typicallyincluded:

## USECASE DIAGRAM

The Use Case Diagram for the Faculty Management System illustrates the interactions between various users (Faculty Members, HODs, and Admins) and the system. It depicts several use cases such as "Update Profile," "View Profile," "Manage Faculty," and "Generate Reports."



**Fig 4.2.1.1 Use Case diagram**

### Explanation

#### Actors:

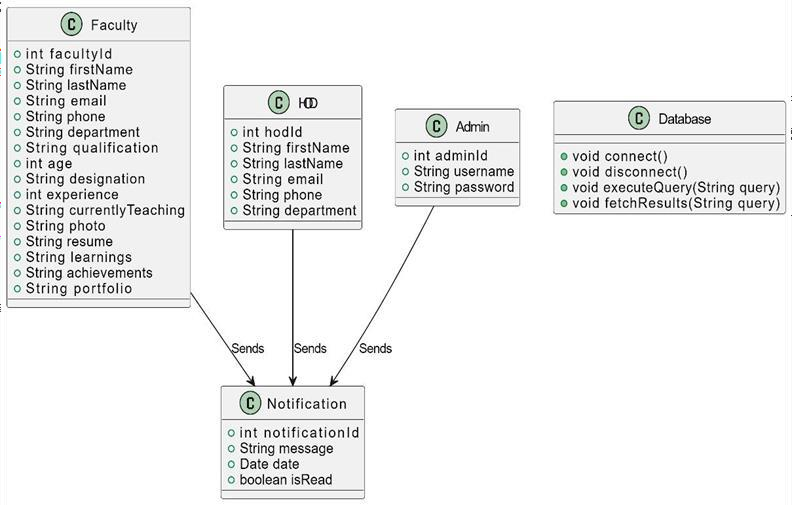
* **Faculty Member:** Represents the faculty users who update their profiles, view their information, and manage their details.
* **HOD (Head of Department):** Represents the department heads who manage faculty data, filter faculty information, and view reports.
* **Admin:** Represents administrative users who handle overall management, generate reports, and maintain systemintegrity.

#### Use Cases:

* **Update Profile:** Allows faculty members to enter or update theirpersonal and professional information in the system.
* **View Profile:** Enables faculty members and HODs to view thedetails of a specific faculty member's profile.
* **Manage Faculty:** Allows HODs and Admins to view, filter, andmanage the list of faculty members within their departments.
* **Generate Reports:** Enables Admins to generate reports based on faculty data, including performance metrics and departmental statistics.

## CLASS DIAGRAM

The Class Diagram for the Faculty Management System describes the static structure of the system, showcasing the classes, their attributes, and therelationships between them. Key classes include Faculty, HOD, Admin, and Course.



**Fig 4.2.2.1 Class diagram**

### Explanation Classes:

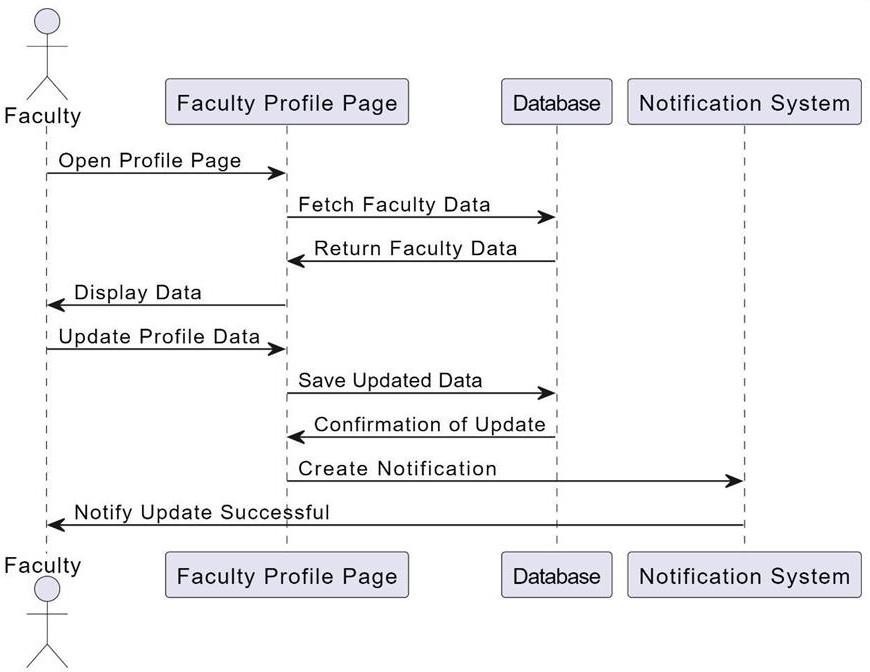
* **Faculty:** Represents individual faculty members using the system. Attributes include FacultyID, FirstName, LastName, Email, PasswordHash, Phone, Department, Qualification, Experience, and ProfilePhoto.
* **HOD (Head of Department):** Represents the heads of departments. Attributes include HODID, FirstName, LastName, Email, PasswordHash, Department, and ContactInfo.
* **Admin:** Represents administrative users managing the system. Attributes include AdminID, Username, PasswordHash, and Role.
* **Course:** Represents the courses offered by the institution. Attributesinclude CourseID, CourseName, Credits, Department, andCurrentlyTeaching.

### Relationships:

* **Faculty "1" -- "0..\*" Course:** A faculty member can teach multiplecourses, but each course is assigned to one faculty member.
* **HOD "1" -- "0..\*" Faculty:** A HOD can oversee multiple faculty members, while each faculty member reports to one HOD.
* **Admin "1" -- "0..\*" Faculty:** An admin can manage multiple faculty members, with each faculty member being handled by one admin.
* **Course "0..*"* -- "0.." Faculty:** A course can have multiple faculty members, and a faculty member can teach multiple courses, establishing a many-to-many relationship.

This Class Diagram provides a clear overview of the entities within the Faculty Management System and their interactions, ensuring organized data handling and efficient management of faculty and courses.

## SEQUENCE DIAGRAM

The Sequence Diagram for the Faculty Management System illustrates the sequence of interactions between objects in the system during specific operations, such as updating a faculty profile or filtering faculty members.

**Fig 4.2.3.1 Sequence diagram**

### Explanation

#### Actors and Participants:

* **Faculty:** Represents the user who interacts with the system to update their profile or filter faculty data.
* **Faculty Management System:** Represents the application processing the faculty operations.
* **Database:** Represents the database where faculty and related data are stored.
* **Notification Service:** Represents the service that sends outconfirmation or status notifications.

#### Interactions:

* **Faculty - System:** The faculty member submits their updated profile information or filtering criteria through theFaculty Management System.
* **System - DB:** The Faculty Management System stores theupdated profile details or executes a filter query in the database.
* **DB - System:** The database returns the necessary confirmation or filtered data back to the Faculty Management System.
* **System - Notification Service:** The system sends a notification to the notification service to inform the faculty member about the successful update or filtering operation.
* **Notification Service - Faculty:** The notification service sends a confirmation message or status update to the faculty member.
* **System - Faculty:** The Faculty Management System confirms that the profile has been successfully updated or presents the filtered results to the faculty member.

This Sequence Diagram provides a detailed view of the interactions involved in key operations within the Faculty Management System, ensuring clarity in how data flows and is processed within the system like- this sequence of interactions is integral to the FMS, ensuring not only the accuracy and security of the data handled but also providing real- time feedback to the users. The system's ability to notify faculty memberspromptly after an operation also enhances the overall user experience by maintaining transparency and communication at every step of the process.

This structured workflow not only guarantees a smooth operation but also prioritizes user communication and satisfaction. By incorporating both real- time notifications and on- screen confirmations, the system enhances its reliability and transparency. Additionally, by integrating security measures during database interactions, the FMS ensures that all sensitive data is handled with care and accuracy, safeguarding the trust of its users.

## SYSTEM DESIGN

* + 1. **MODULAR DESIGN**

Modular design in system architecture involves dividing the system into distinct, self- contained modules, each responsible for a specific function. In the Faculty Management System, separate modules handle faculty profiles, HOD management, admin control, and login authentication. This approach ensures each module can be developed, tested, and maintained independently, improving flexibility and scalability. Modular design allows easier updates or changes to individual parts without impacting the entire system. It also enhances code reusability and simplifies debugging, makingthe system more efficient and maintainable.

## DATABASE DESIGN

The database design of the Faculty Management System is structured to efficiently store and manage the data of faculty members, Heads of Departments (HODs), and administrators. The system utilizes a MySQL relational database, which is divided into several tables to handle the various aspects of faculty management. The primary tables include faculty, hods, and admins. Each table serves a specific purpose, with the faculty table holding individual faculty information such as their first name, last name, age, qualification, department, date of birth (dob), date of joining (doj), email, phone number, aadhar number, specialization, designation, currently teaching subjects, experience, and university.

The hods table stores details related to the department heads, including their name, qualifications, department, and experience, while the admins table manages administrator credentials and roles. Each of these tables has primary keys that uniquely identify each record and are related to foreign keys where necessary to establish relationships between tables. For instance, the faculty table may have a foreign key referencing the hods table, associating faculty members with their respective department heads.

To ensure data integrity and smooth data management, the database design incorporates constraints, such as NOT NULL, UNIQUE, and foreign key constraints, ensuring that every record entered is complete and that relationships between tables remain consistent. The design also includes appropriate indexes to speed up queries related to filtering faculty based on parameters such as age, experience, and department.

The system supports a range of user roles (faculty, HOD, and admin), eachwith different levels of access and privileges. Faculty members can update their profiles, upload documents like resumes, and enter their work experience, qualifications, and course details. HODs, on the other hand, have the capability to filter faculty based on criteria like specialization, age, experience, and department, and can view or manage the details of faculty members within their department. Admins manage the overall system, including adding new faculty members or HODs, as well as ensuring system integrity.

The design also supports document storage for faculty members, allowing them to upload resumes, certificates, and other important documents. These documents are stored in the system and referenced in the faculty table, providing quick access to all related materials when needed. Additionally, there is functionality for viewing and downloading uploaded files, enablingeasy management of faculty portfolios.

To handle various actions like filtering faculty data or updating profiles, thedatabase design includes efficient querying mechanisms. This ensures that HODs can filter faculty details based on input criteria quickly and accurately, making use of indexes on columns such as age, experience, and specialization. The filtering system relies on SQL queries that are dynamically generated based on the user's input, ensuring flexibility and accuracy in the results.

In summary, the database design of the Faculty Management System is robust, secure, and optimized for handling a variety of user roles and data interactions. It ensures that the faculty data is efficiently stored, retrieved, and managed while maintaining relationships between various entities like faculty members, HODs, and admins. The design is scalable, allowing for future expansion and adaptation to meet the growing needs of the institution.

# IMPLEMENTATION

## IMPLEMENTATION

The implementation of the Faculty Management System is divided into several key sections to ensure modularity, scalability, and ease of use. This section outlines the process and considerations during the project's development.

#### Backend Development

The backend of the system is implemented using PHP, which interacts witha MySQL database to handle data processing and storage. The backend manages user authentication, data retrieval, and manipulation, ensuringsecure and efficient access to faculty, HOD, and admin information. Each role in the system—admin, faculty, and HOD—has specific privileges, and backend logic ensures that users can only access data relevant to their role.

#### Frontend Development

The user interface (UI) is created using HTML, CSS to provide an intuitive and user- friendly experience. The front end allows faculty members, HODs,and admins to log in, view, update, or manage profiles and filter data as pertheir needs. Bootstrap is also used to make the interface responsive, ensuring compatibility across different devices, including desktops, tablets, and mobile phones.

#### Database Integration

The MySQL database is integrated with PHP backend to store, retrieve, andmanage all system data. This includes faculty profiles, department details, and admin credentials. SQL queries are used to handle complex filtering tasks, ensuring fast and accurate data retrieval. Passwords are encrypted for security, and foreign key relationships are maintained to ensure data integrityacross multiple tables.

#### User Authentication and Authorization

A key aspect of the system is the authentication mechanism, which ensures that only authorized users can access the system. Admins, HODs, and faculty members have separate login interfaces, with their credentials verified against the database. Role- based authorization ensures that each user can only access and modify data pertinent to their role.

#### Profile Management

Faculty members are able to manage their profiles through an easy-to-use interface. They can upload documents like resumes and certificates, and update fields such as qualifications, experience, and courses they are currently teaching. Once updated, the data is stored in the faculty table in thedatabase, and admins or HODs can view or filter these details.

#### Data Filtering and Reporting

HODs can filter faculty members based on various criteria, such as age, experience, and specialization. The filtering functionality is implemented using dynamic SQL queries that retrieve data based on user inputs. Results are displayed in a tabular format and can also be downloaded as Excel files for reporting purposes. This feature helps streamline departmental management by providing filtered views of faculty members.

#### Testing and Validation

The system was thoroughly tested for functionality, usability, and performance. Unit tests were performed for each module, including login, profile management, and filtering. Validation checks were implemented to ensure that all required data fields are completed accurately and securely before any data is submitted to the database.

#### Deployment

After successful development and testing, the Faculty Management System was deployed on a local server using XAMPP. This allows the system to be accessed through a browser interface, making it available to faculty, HODs, and admins within the institution's network. Future plans include hosting the system on a cloud-based server to facilitate remote access.

#### Security Measures

The system incorporates multiple layers of security to protect sensitive data. Passwords are hashed using secure algorithms, and sessions are managed to prevent unauthorized access. SQL injection attacks are mitigated by using prepared statements in all database queries. In addition, access logs track user activity, providing an audit trail that can be used to monitor for any unusual or malicious behavior. The system also implements role-based access control, ensuring that only users with the appropriate permissions can view or edit certain sections of the database.

#### Scalability and Future Enhancements

The system is designed with scalability in mind, allowing for future enhancements as the institution grows. The modular architecture makes it easy to integrate new features, such as additional user roles or extended reporting capabilities. Future updates may include integrating third-party APIs for automated data synchronization and enhancing the front-end with more advanced frameworks like React or Vue.js for a more dynamic user experience. Additionally, plans are in place to migrate the system to a cloud-based infrastructure, which will offer greater flexibility, performance, and remote access capabilities for users.

The implementation of the Faculty Management System emphasizes security, performance, and scalability, ensuring a robust solution for managing faculty profiles and departmental operations.

* 1. **SAMPLE CODE**

#### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,initial-scale=1.0">

<title>Faculty Management System</title>

<link rel="stylesheet" href="css/styles.css">

<style>

/\* General Styles \*/body {

font-family: Arial, sans-serif;margin: 0; padding: 0;

background-color: #f4f4f9;

}

.container {

text-align: center; padding: 50px 20px;

}

h1 {

font-size: 36px;color: #333; margin-bottom: 20px;

}

ul {

list-style: none;padding: 0;

}

ul li {

display: inline-block;margin: 0 15px;

}

ul li a {

display: inline-block; padding: 15px 30px; background-color: #007bff;color: #fff; text-decoration: none;font-size: 18px; border-radius: 5px;

transition: background-color 0.3s ease;

}

ul li a:hover {

background-color: #0056b3;

}

/\* Responsive Design \*/ @media (max-width: 768px) { ul li {

display: block; margin: 10px 0;

}

ul li a {

padding: 12px 20px;font-size: 16px;

}

}

/\* Footer \*/footer {

margin-top: 30px;font-size: 14px; color: #777;

}

</style>

</head>

<body>

<div class="container">

<h1>Welcome to Faculty Management System</h1>

<ul>

<li><a href="admin/admin\_login.php">Admin</a></li>

<li><a href="faculty/faculty\_login.php">Faculty</a></li>

<li><a href="hods/hods\_login.php">HODs</a></li>

<li><a href="about\_us.html">About Us</a></li>

</ul>

</div>

<footer>

<p>&copy; 2024 Faculty Management System. All RightsReserved.</p>

</footer>

</body>

</html>

# TESTING

## TESTING

Testing is a critical phase in the development of the Faculty Management System, ensuring that the application functions correctly and meets user requirements. Various testing methodologies are employed throughout the project lifecycle to identify and rectify issues before deployment.

* Unit Testing: This involves testing individual components or modules of the system, such as user authentication, profile updates, and filtering functionalities. Each unit is tested in isolation to verify its correctness, ensuring that it behaves as expected. Automated testing frameworks can be employed to streamline this process.
* Integration Testing: After unit testing, the integration of various modules is tested to ensure that they work together seamlessly. For instance, the interaction between the faculty profile module and thedatabase is assessed to confirm that data is stored and retrieved accurately. This helps in identifying issues related to data flow between modules.
* Functional Testing: This type of testing focuses on evaluating the system against specified functional requirements. Test cases are created to validate that each feature, such as the login process, profile management, and filtering capabilities, works as intended. User acceptance testing (UAT) is often part of this phase, involvingactual users to validate the system's functionality.
* Performance Testing: Performance testing assesses the system's responsiveness, stability, and scalability under various load conditions. It ensures that the Faculty Management System can handle multiple simultaneous users without significant slowdowns. Stress testing and load testing are conducted to evaluate performance metrics.
* Security Testing: Security is a paramount concern in any management system. Security testing involves identifyingvulnerabilities and ensuring that sensitive data, such as facultyinformation and credentials, is protected. Testing for SQL injection,

cross-site scripting (XSS), and other common threats is conducted to ensure robust security measures.

* Regression Testing: As new features are added or changes are made,regression testing ensures that existing functionalities remain unaffected. Automated tests can be run to quickly verify that previous code changes do not introduce new bugs.

#### Conclusion:

The testing phase plays a critical role in ensuring the overall success of the Faculty Management System (FMS). It is during this phase that the functionality, performance, and security of the system are rigorouslyvalidated. Comprehensive testing of every component within the system allows developers to identify and rectify any potential issues before the system is deployed in a live environment. This includes validating the system’s core features such as user registration, faculty profile management, HOD assignments, and admin control functionalities, all of which are essential to the system's day-to-day operations.

Thorough testing ensures that the system performs seamlesslyacross different use cases, from routine login procedures to more complex administrative tasks. By simulating various real-world scenarios, the developers can evaluate how well the system handles expected loads, user interactions, and security challenges such as unauthorized access attempts or system vulnerabilities. This helps in mitigating any risks that could potentially compromise the integrity of the system or the privacy of user data.

In conclusion, the testing phase is much more than a technical requirement; it is an essential process that ensures Faculty Management System operates as intended and meets the needs of its diverse user base. Bythoroughly addressing all aspects of the system during this phase, developers are able to deliver a high-quality, secure, and efficient product, thereby contributing to the long-term success of the project. This not only elevates overall user experience but also establishes a strong foundation for futuresystem enhancements and updates. Ultimately, successful testing transforms the system from a collection of features into a cohesive, reliable solution thatfaculty and administrators can trust.

**2 TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Input | Expected Output | Actual Output | Pass/ Fail |
| TC001 | User Registration | Valid user details (Admin/ Faculty/ HOD) | User account is created successfully | As expected | Pass |
| TC002 | User Registration | Missing requiredfields | Error message: 'All fields are required' | As expected | Pass |
| TC003 | User Login | Valid login credentials | User is logged in and redirected to the dashboard | As expected | Pass |
| TC004 | User Login | Invalid login credentials | Error message: 'All fields are required' | As expected | Pass |
| TC005 | Faculty Profile Update | Valid profile details | Profile is updated successfully | As expected | Pass |
| TC006 | Faculty Profile Update | Missing requiredfields | Error message: 'All fields are required' | As expected | Pass |
| TC007 | Faculty Management (Add Faculty) | Valid faculty details (Admin/HOD role) | New faculty is added to the system | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC008 | Faculty Management (Delete Faculty) | Valid faculty ID | Faculty is successfully removed from the system | As expected | Pass |
| TC009 | Faculty Management (Edit Faculty) | Valid faculty details for update | Faculty details are updated  successfully | As expected | Pass |
| TC010 | View Faculty Records | Authorized user(HOD/ Admin) | Facult y Records are  displayed correctly | As expected | Pass |
| TC011 | View Faculty Records | Unauthorized user | Error message: 'Access Denied’ | As expected | Pass |
| TC012 | Database Connectivity | System boot-up | Successful connection to database | As expected | Pass |
| TC013 | Database Connectivity | Network failure during operation | Error message: | As expected | Pass |
| TC014 | System Performance | High number of simultaneous users | System handles load without performance drop | As expected | Pass |
| TC015 | Logout Functionality | Logged-in user | User is logged out and  redirected to the login page | As expected | Pass |
| TC016 | Password Management | Valid password reset request | Password reset link is sent successfully | As expected | Pass |

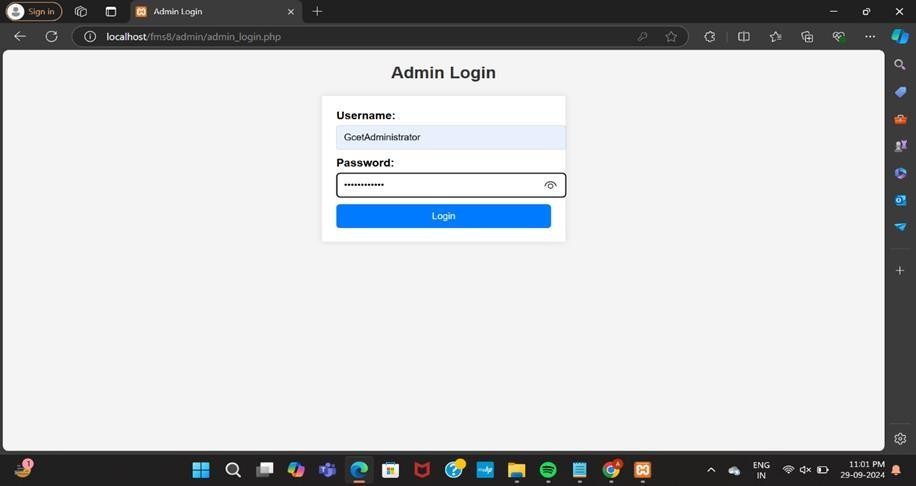
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC017 | Password Management | Invalid password reset request(e.g., unregistered  email) | Error message: 'Email not found' | As expected | Pass |
| TC018 | Faculty Assignment (HOD Role) | Valid faculty assignment details | Faculty is assigned to a department successfully | As expected | Pass |
| TC019 | View Reports (Admin) | Valid faculty activity records | Reports are generated and displayed correctly | As expected | Pass |
| TC020 | Faculty Search | Valid search criteria (e.g., faculty name) | Relevant faculty records are displayed | As expected | Pass |

# OUTPUTS

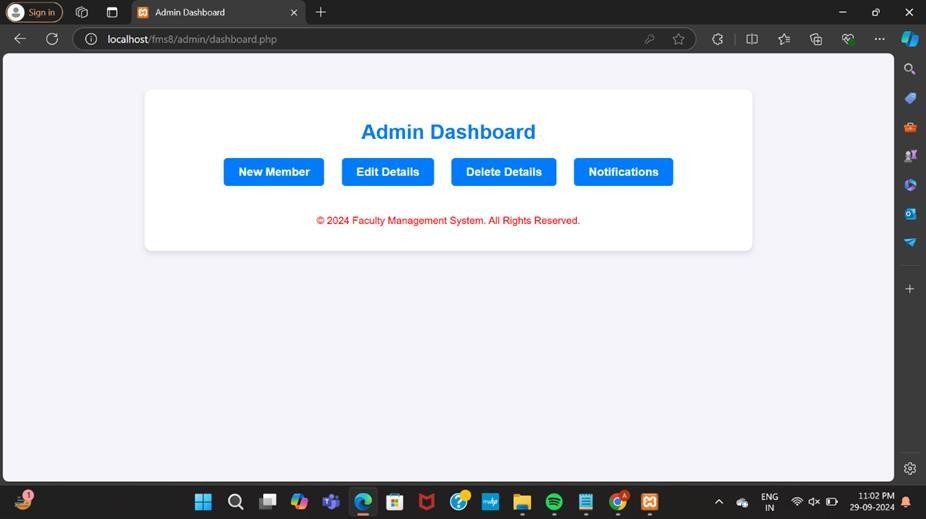
## WEB INTERFACE



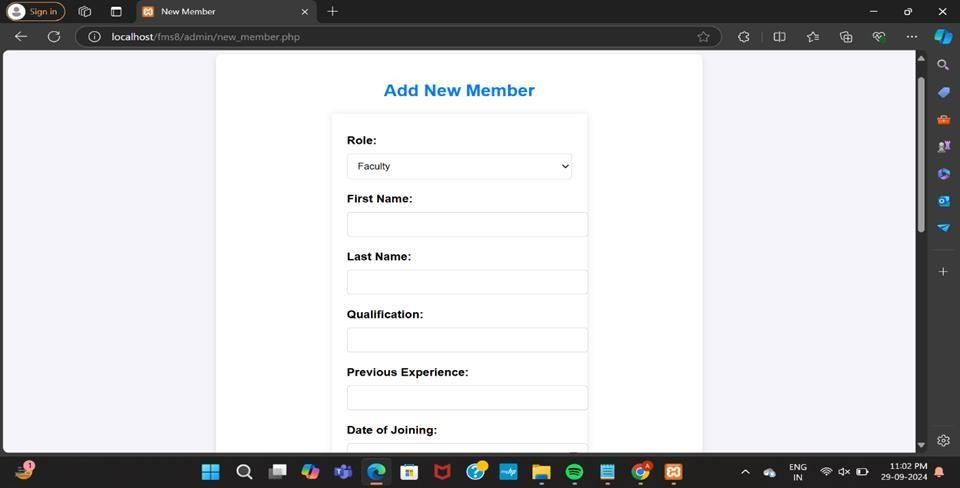
**Fig 7.1 Main Interface of the faculty management system**



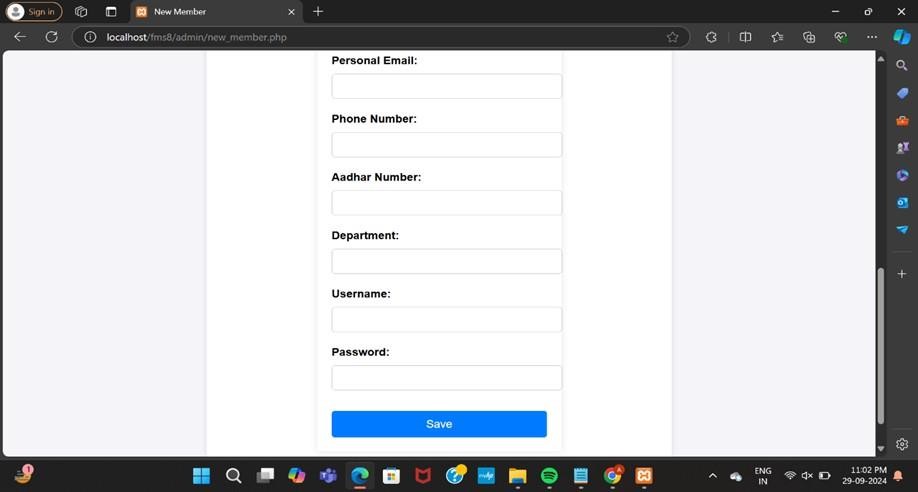
**Fig 7.2 Admin Login Screen**



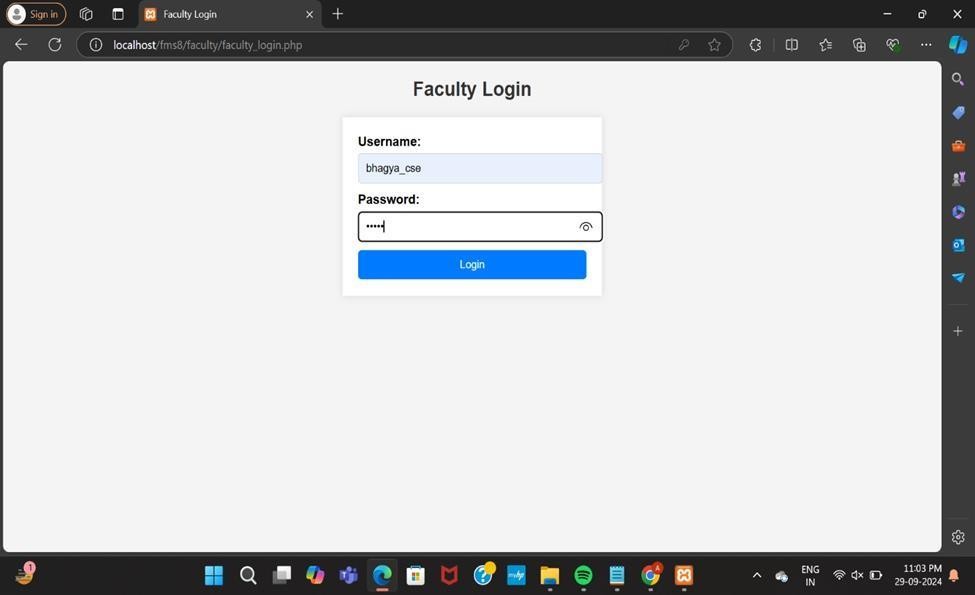
**Fig 7.3 Admin Dashboard**



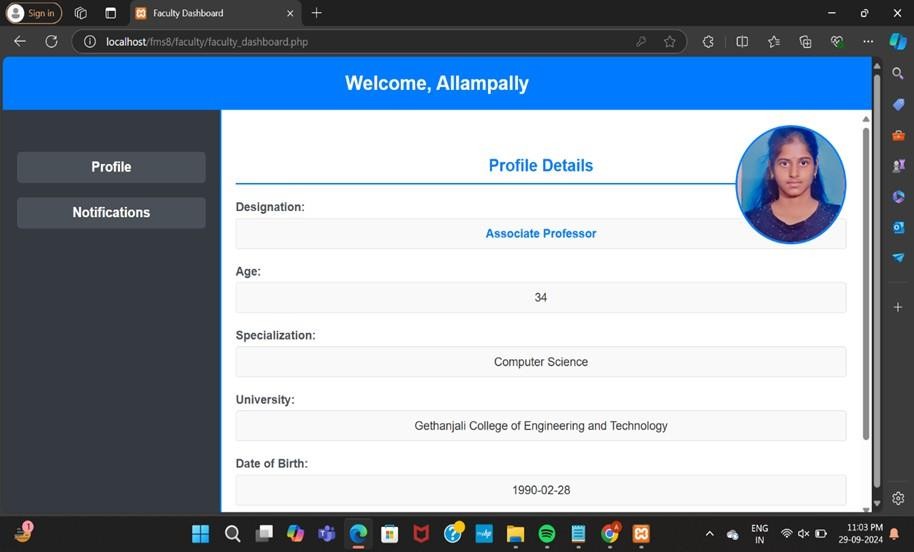
**Fig 7.4 Add new member screen**



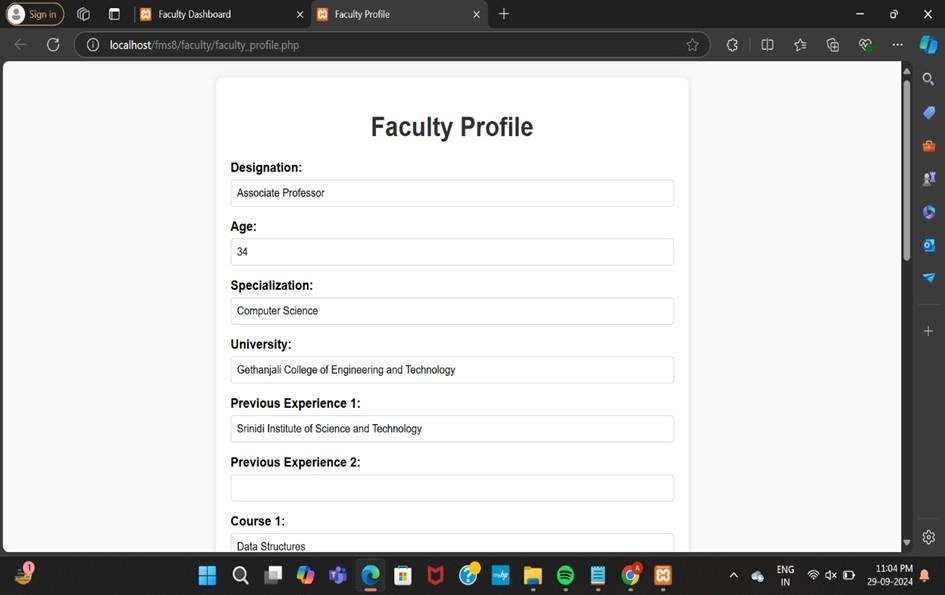
**Fig 7.5 Save new member**



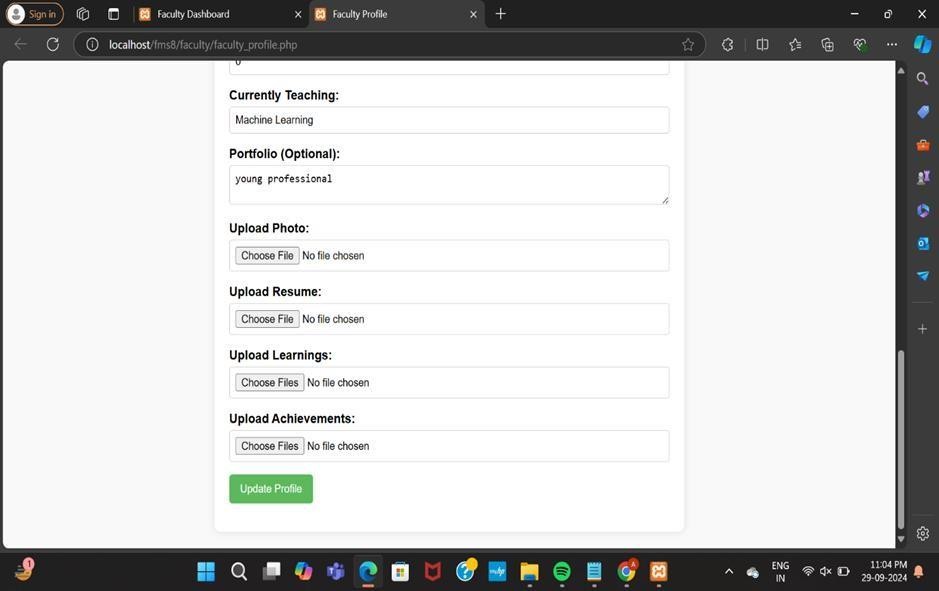
**Fig 7.6 Faculty login**



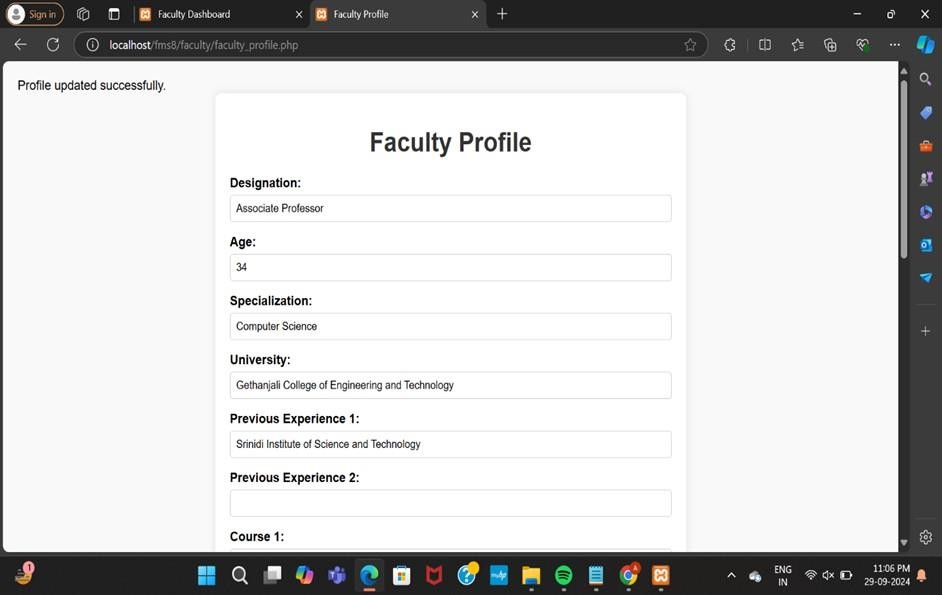
**Fig 7.7 Faculty dashboard**



**Fig 7.8 Faculty profile Page**



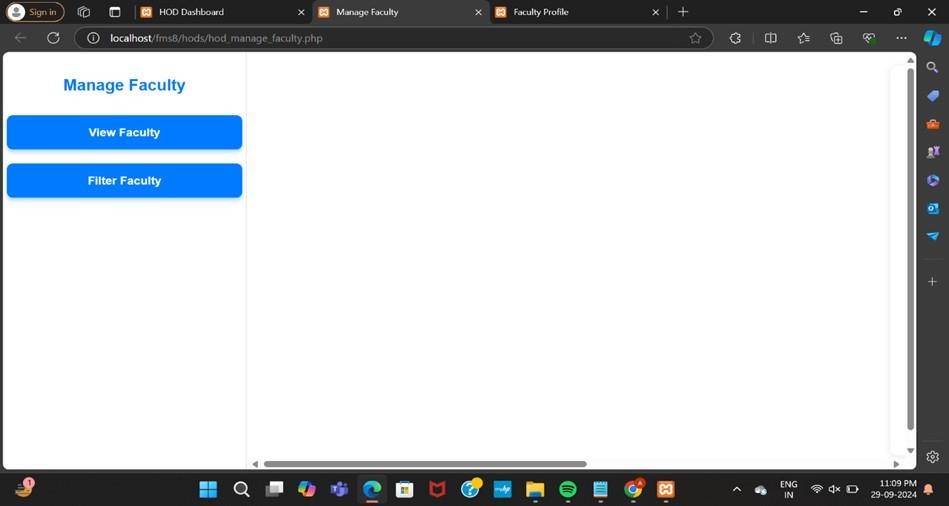
**Fig 7.9 Faculty profile update page**



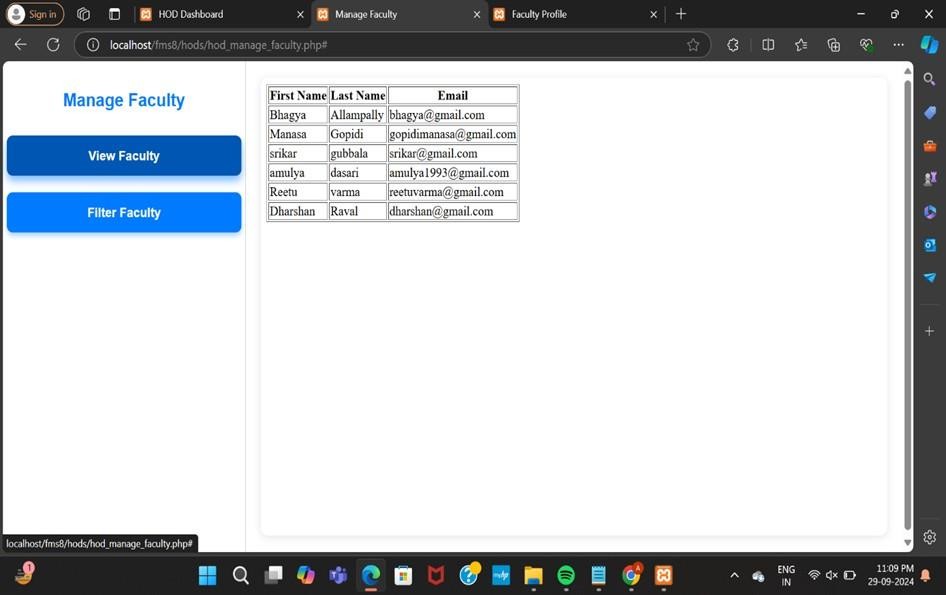
**Fig 7.10 Post update page**



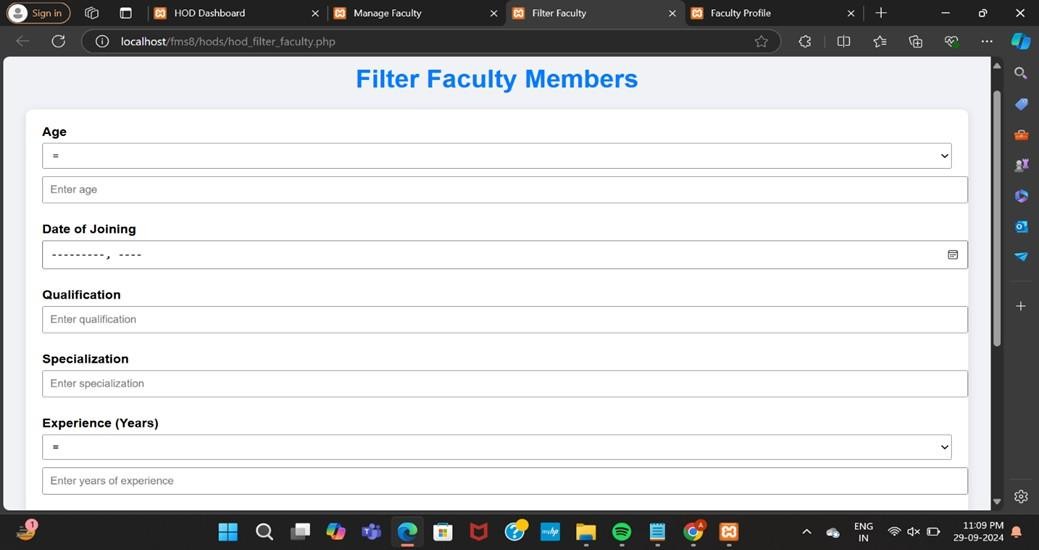
**Fig 7.11 Hods Dashboard**



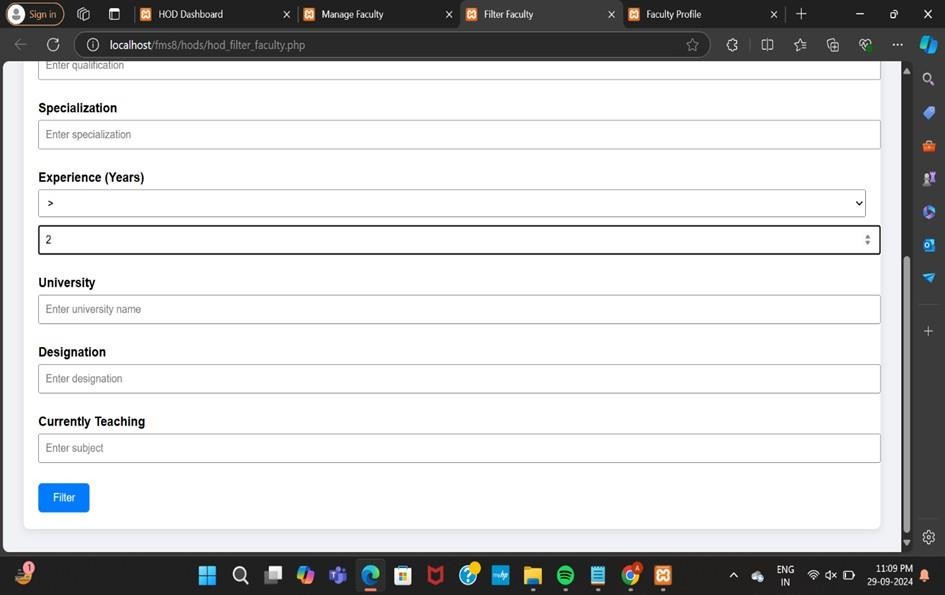
**Fig 7.12 Manage Faculty page**



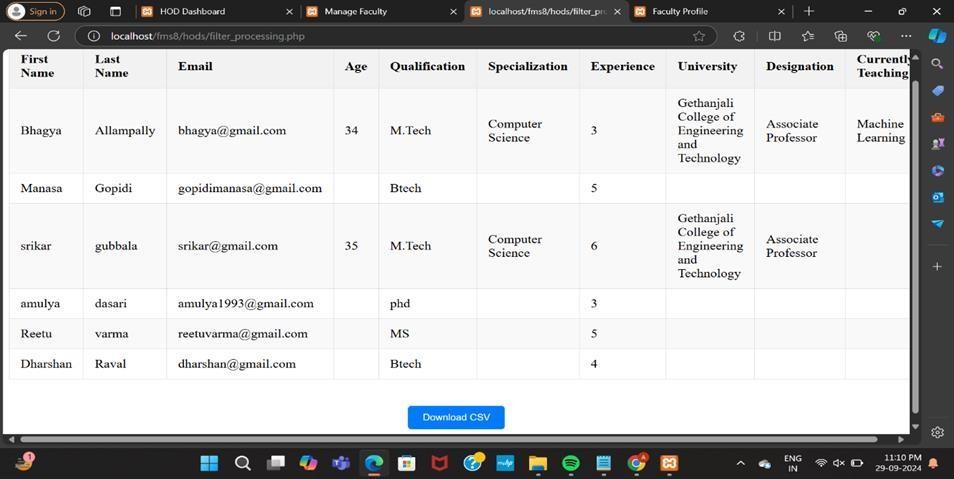
**Fig 7.13 View Faculty page**



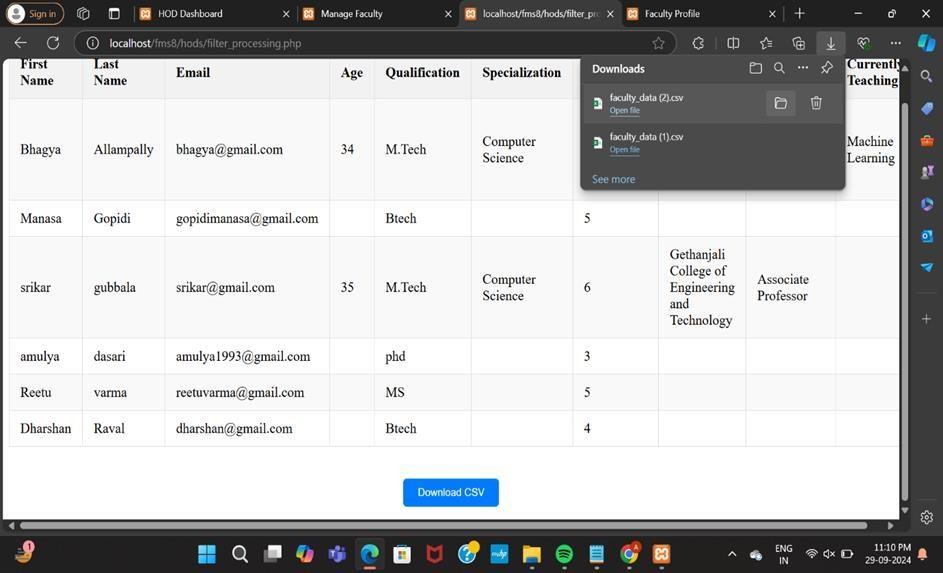
**Fig 7.14 Filter Faculty page**



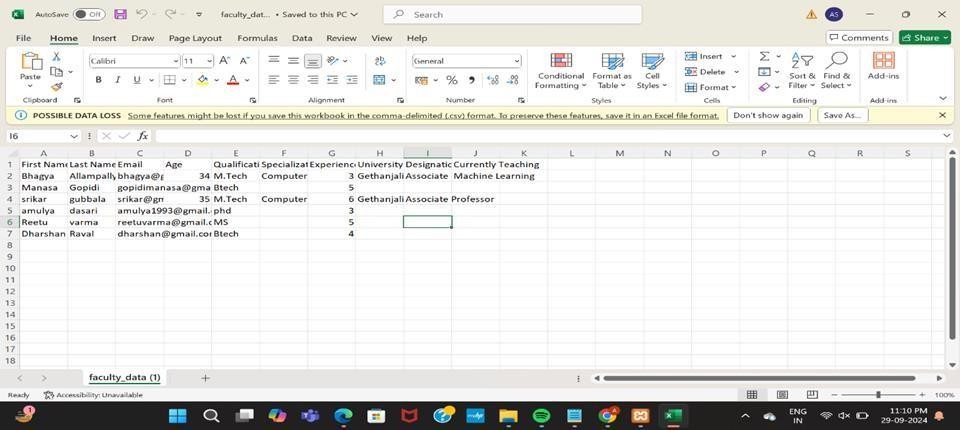
**Fig 7.15 Filtering Faculty page**



**Fig 7.16 Filtered Faculty page**



**Fig 7.17 Download Excel option page**



**Fig 7.18 Downloaded Excel page**

# CONCLUSION

## CONCLUSION

The Faculty Management System represents a significant advancement in the management and administration of educational institutions. By harnessing modern web technologies, the system provides a streamlined, efficient, and user-friendly platform for faculty members, heads of departments, and administrators to manage academic and operational tasks seamlessly.

Traditionally, faculty management involved some paperwork, manual record-keeping, and inefficient communication channels, which could hinder the productivity of both educators and administrators. The Faculty Management System eliminates thesechallenges, allowing faculty members to access their profiles, update their information, and interact with departmental resources from any location with internet access. This ease of use is particularly valuable in promoting timelyupdates and enhancing communication between faculty and administration.

In addition to simplifying tasks for faculty, the system significantly improves the operational efficiency of educational institutions. By digitizing essential functions such as profile management, notifications,and faculty filtering, the Faculty Management System reduces administrative workloads, enabling staff to focus on enhancing educational outcomes. The electronic storage and retrieval of faculty records ensure that vital information is always accessible, which aids in making informed decisions related to faculty assignments, course offerings, and resource allocations.

Moreover, the Faculty Management System fosters transparency and accountability within the institution. Faculty members can track their profiles, view notifications, and access resources in real time, ensuring they are informed and engaged in their academic environment. This transparencystrengthens trust between faculty and administration, which is crucial for fostering a collaborative and supportive educational atmosphere.

In conclusion, the Faculty Management System is more than just a technological advancement; it is a step towards a more organized, transparent, and effective educational environment. By bridging the gap between faculty and administration, the system contributes to the creation of a more supportive and accountable academic framework, ultimately enhancing the quality of education and fostering a vibrant learning community.

## FURTHER ENHANCEMENTS

While the Faculty Management System is a comprehensive solution, several potential enhancements could further improve itsfunctionality:

* **Mobile Application Development**: Creating a mobile app version of the Faculty Management System would enhanceaccessibility, enabling faculty members and administrators to manage profiles, view notifications, and communicate on-the- go.
* **Advanced Analytics Integration**: Incorporating data analytics and reporting features could help institutions track faculty performance, analyze trends in course offerings, and improve decision-making related toresource allocation.
* **Multilingual Support**: Adding support for multiple languages would ensure the system is more inclusive, catering to a diverse faculty population and facilitating better communication.
* **Integration with Academic Databases**: Linking the Faculty Management System with national or regional academic databases could provide access to additional resources, such as research publications and professional development opportunities, enhancing faculty engagement.
* **Real-Time Notifications**: Implementing real-timenotifications for updates on profile changes, departmental announcements, and important deadlines would keep faculty membersinformed and engaged with their academic environment.
* **Enhanced Security Features**: Continuous improvement ofsecurity protocols, such as two-factor authentication and data encryption, will ensure the protection of sensitive information and compliance with data privacy regulations.

# BIBILOGRAPHY

## BOOKS REFERENCES

1. **Sommerville, Ian.** *Software Engineering*, 10th Edition, Pearson, 2015.
2. **Korth, Henry F., and Silberschatz, Abraham.** *Database System Concepts*, 6th Edition, McGraw-Hill, 2011.
3. **Schneider, David I., and Davis, R.** *An Introduction to Programming Using PHP*, Pearson, 2013.

## WEBSITE REFERNCES

1. **W3Schools.** *HTML Tutorial*[,](http://www.w3schools.com/html/) [www.w3schools.com/html](http://www.w3schools.com/html)
2. **PHP Documentation.** *PHP: Hypertext Preprocessor*[,www.php.net/docs.php](http://www.php.net/docs.php)
3. **XAMPP Documentation.** *XAMPP: Easy Apache Distribution*[,](http://www.apachefriends.org/) [www.apachefriends.org](http://www.apachefriends.org/)

## TECHNICAL PUBLICATION REFERENCES

1. **Sharma, A., and Verma, R.** "Design and Implementation of a Faculty Management System Using PHP and MySQL," *International Journal of Computer Applications*, Vol. 88, No. 12, pp. 18-22, 2014.
2. **Patel, M., and Singh, A.** "A Secure Web-Based Faculty Management System," *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018.
3. **Kumar, S., and Gupta, R.** "Web-Based Systems for Managing Educational Institutions: Challenges and Solutions," *Journal of Software Engineering and Applications*, Vol. 11, No. 5, pp. 150-158, 2019.

This bibliography covers a comprehensive range of references that support both the technical and theoretical aspects of the FMS project.

# APPENDICES

## APPENDIX A- SOFTWARE USED

#### XAMPP

* + - * **Purpose**: XAMPP was used as the local development environment for the Faculty Management System. It provided the necessary components such as Apache (web server), MySQL (database), and PHP (server-side scripting language).
      * **Version**: XAMPP v7.4.2

#### PHP

* + - * **Purpose**: PHP was used as the primary server-side scriptinglanguage to handle the business logic of Faculty Management System, including form processing, database interactions, and session management.
      * **Version**: PHP v7.4

#### MySQL

* + - * **Purpose**: MySQL served as the database management system, wherefaculty records, user information, and course details were securelystored.
      * **Version**: MySQL v5.7

#### HTML/CSS

* + - * **Purpose**: HTML and CSS were used for the front-end design of the Faculty Management System, creating the user interface that allows faculty members, HODs, and administrators to interact with the system.
      * **Tools**: Various text editors and IDEs, such as Visual Studio Code, were used to develop the front-end code.

## APPENDIX B- METHODOLOGIES USED

#### Agile Methodology

* + - * **Purpose:** The Agile methodology was employed to ensure iterative development of the Faculty Management System. This approach allowed for continuous feedback and improvement, with regular sprints focusing on different modules such as user login, profile management, faculty filtering, and data export.

#### Modular Development

* + - * **Purpose:** The Faculty Management System was developed in a modular fashion, with distinct modules for user authentication, faculty profile management, data filtering, and notification handling. This approach enhanced maintainability and allowed for easier testing and debugging.

#### MVC Architecture

* + - * **Purpose:** The Model-View-Controller (MVC) architecture was implemented to separate the application logic, user interface, and data management. This separation of concerns made the system more organized, scalable, and maintainable, facilitating easy updates to different sections such as profile pages or dashboard functionality.

## APPENDIX C - SYSTEM ARCHITECHTURE

#### Unit Testing

* + - * **Purpose**: Each module and function within the Faculty Management System was tested individually to ensure that they performed as expected. Unit tests were particularly focused on validating input forms, profile updates, login processes, and database operations.

#### Integration Testing

* + - * **Purpose**: After individual modules were tested, integration testing was conducted to verify that these modules interacted correctly with each other. This ensured seamless communication between the database, server-side

scripts, and user interface components like the faculty profile pages and dashboard.

#### System Testing

* + - * **Purpose**: The entire system was tested as a whole in the deployment environment to check for overall functionality. This phase included testing the complete workflow from user login, profile updates, to datafiltering and report generation.

#### User Acceptance Testing (UAT)

* + - * **Purpose:** Potential end-users, including faculty members, HODs, and administrators, were involved in testing the system. Their feedback was crucial in ensuring that the system was user-friendly and met practical requirements, such as data accuracy, profile navigation, and ease of use in managing faculty information.

#### Security Testing

* + - * **Purpose:** Security testing was performed to protect sensitive user data and prevent unauthorized access. This included testing the strength of user authentication mechanisms, secure data