

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
BELAGAVI**



A Project Report on

**LEARN SYNC**

Submitted in partial fulfillment of the requirement for the award  
of the degree of

**MASTER OF COMPUTER APPLICATIONS**

Under

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

By

**SUSHMITHA MONTHERO**

**4SO23MC107**



**Department of Computer Applications  
St Joseph Engineering College  
Mangaluru-575028**

**2025**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
BELAGAVI**



A Project Report On

**LEARN SYNC**

Submitted in partial fulfillment of the requirement for the award of the  
degree of

**MASTER OF COMPUTER APPLICATIONS**

Under

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

By

**SUSHMITHA MONTHERO**

**4SO23MC107**

Under the Guidance of

**Dr. Gururaja S**

Assistant Professor

Department of Computer Applications

St Joseph Engineering College Mangaluru-575 028.



**Department of Computer Applications  
St Joseph Engineering College  
Mangaluru-575028**

**2025**

ST JOSEPH ENGINEERING COLLEGE,  
MANGALURU  
DEPARTMENT OF COMPUTER APPLICATIONS



**CERTIFICATE**

*This is to certify that the project work titled*

**LEARN SYNC  
SUBMITTED BY**

**SUSHMITHA MONTHERO  
4SO23MC107**

*In partial fulfillment of the requirements for award of degree of Master of Computer Applications of Visvesvaraya Technological University, is a bonafide record of the work carried out*

*During the academic year  
2024-2025.*

Dr. Gururaja S 11/8/25

Assistant Professor  
Department of Computer Applications  
St Joseph Engineering College  
Mangaluru-575 028.

Dr Hareesh B 01/08/25

HOD-MCA  
St Joseph Engineering College  
Mangaluru-575 028.

Examiner 1.

11/08/2025

Ms. Priyadarshini P  
Project Coordinator  
Department of Computer Applications  
St Joseph Engineering College  
Mangaluru-575 028

Dr Rio D'Souza

Principal  
St Joseph Engineering College  
Mangaluru-575 028.

Examiner 2.

8. Ga  
11/8/2025

## **DECLARATION**

I hereby declare that the entire work embodied in this dissertation has been carried out by me and no part of it has been submitted for any degree or diploma of any university previously.

Place: Mangalore

Date:

Name: Sushmitha Monthero

USN: 4SO23MC107

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all those who supported and guided me throughout the development of this project titled Learn Sync undertaken as part of my final semester curriculum at St. Joseph Engineering College, Mangaluru.

I am deeply thankful to the Department of Computer Applications for providing a strong academic foundation and a technically supportive environment that enabled me to pursue this project successfully. I express my heartfelt thanks to Dr. Rio D'Souza, Principal of St. Joseph Engineering College, for providing the institutional support and infrastructure required for the smooth execution of the project.

I would also like to sincerely thank Rev. Fr. Wilfred Prakash D'Souza, Director of SJEC, and the Management of SJEC for fostering a learning culture that encourages student-led innovation and research.

My special thanks go to Dr. Hareesh B, Head of the Department of Computer Applications, for his encouragement, consistent academic support, and valuable suggestions throughout the project.

I am grateful to Assistant Professor Dr. Gururaja S, my project guide, for his expert guidance, constructive feedback, and dedicated mentorship, which were instrumental in shaping and successfully completing this project.

I extend my thanks to our Project Coordinator, Assistant Professor Ms. Priyadarshini P, for her coordination, timely reviews, and support throughout the project timeline.

I also acknowledge the unwavering support of the faculty members and lab staff of the department for facilitating a productive and technically equipped environment. Their cooperation was invaluable, especially during the implementation and testing stages.

Lastly, I would like to thank my classmates and friends for their encouragement, helpful discussions, and continuous support during this journey.

This project has been a profound learning experience, allowing me to apply my technical knowledge to real-world challenges. I remain truly appreciative of everyone who contributed to its success.

NAME: SUSHMITHA MONTHERO

DATE:

## **ABSTRACT**

The proposed project, Learn Sync, is a web-based academic management platform that aims to simplify and digitize important academic activities. This includes tracking internships, submitting and evaluating projects, uploading MOOC certifications, managing student-faculty mentorships, and overseeing classroom operations (ClassOps). The system is role-based and serves students, faculty, and the head of department (HOD), each with specific access and features. It allows students to submit organized academic records, receive instant feedback on project evaluations, keep track of mentorships, and join digital classrooms, all from a user-friendly interface.

Currently, many institutions depend on manual record-keeping, scattered communication methods, and separate tools to monitor academic progress. This results in inefficiencies with managing student submissions, unclear project approvals, challenges in tracking internships and MOOC certifications, and limited insight into student development beyond grades. Faculty members often find it hard to share mentoring resources and classroom materials due to the lack of a unified system.

Learn Sync tackles these issues by providing a central platform with automated workflows, smart project analysis based on keyword trends, real-time mentorship insights, and features similar to Google Classroom for ClassOps. It replaces paper processes with organized digital ones, ensures secure document uploads, improves feedback systems, and gives stakeholders tailored views and search capabilities. The platform seeks to make academic management more efficient, clear, and ready for the future in educational institutions.

## **TABLE OF CONTENTS**

1. Introduction .....	01
2. Literature Survey .....	02
3. Software Requirements Specification.....	04
4. System Design .....	18
5. Detailed Design.....	29
6. Implementation .....	40
7. Software Testing .....	43
8. Conclusion .....	51
9. Future Enhancements.....	52
Appendix A: References.....	53
Appendix B: User Manual.....	54

## **INTRODUCTION**

Project Description .....	01
---------------------------	----

## **LITERATURE SURVEY**

2.1 Existing and Proposed System .....	02
2.1.1 Existing System .....	02
2.1.2 Proposed System .....	02
2.2 Feasibility Study .....	02
2.3 Tools and Technologies Used .....	03

# **SOFTWARE REQUIREMENTS SPECIFICATION**

3.1 Introduction.....	04
3.1.1 Purpose.....	04
3.1.2 Document Conventions/Definitions and Abbreviations.....	04
3.1.3 Intended Audience and Reading Suggestions.....	05
3.1.4 Project Scope .....	05
3.1.5 Benefits.....	06
3.1.6 References.....	06
3.2 Overall Description.....	07
3.2.1 Identification of Pre-existing work.....	07
3.2.2 Product Perspective.....	07
3.2.3 Product Features.....	07
3.2.4 User Characteristics.....	08
3.2.5 Operating Environment.....	08
3.2.6 Design and Implementation Constraints.....	09
3.2.7 Assumptions and Dependencies .....	09
3.3 Product Functionality.....	10
3.3.1 Module Details.....	10
3.4 External Interface Requirements.....	11
3.4.1 User Interfaces.....	11
3.4.2 Hardware Interfaces.....	12
3.4.3 Software Interfaces .....	12

3.4.4 Communication Interfaces.....	13
3.5 Non-Functional Requirements .....	13
3.5.1 Performance Requirements.....	13
3.5.2 Safety Requirements.....	14
3.5.3 Software Quality Attributes.....	14
3.5.4 Specific Requirements.....	15
3.6 Operating Environment .....	15
3.6.1 Hardware.....	15
3.6.2 Software .....	16
3.7 Delivery Plan .....	16

# **SYSTEM DESIGN**

4.1 Introduction .....	18
4.2 Scope.....	18
4.3 Audience .....	19
4.4 Software Product Architecture.....	21
4.4.1. Architectural Design.....	21
4.4.1.1. View Layer or Presentation Layer.....	21
4.4.1.2. Business Layer .....	22
4.4.1.3. Access Layer .....	22
4.5 Component Architecture.....	23
4.5.1 User Interface.....	24
4.6 Data Flow Diagram.....	24

## **DETAILED DESIGN**

5.1 Use Case Diagram .....	29
5.2 Sequence Diagrams.....	30
5.3 Activity Diagram .....	34
5.4 Database Design.....	36
5.4.1 ER Diagram .....	38

## **IMPLEMENTATION**

6.1 Introduction.....	41
6.2 Pseudo Codes .....	41

# **SOFTWARE TESTING**

7.1 Introduction.....	44
7.2 Testing Object.....	44
7.3 Unit Testing .....	44
7.3.1 Unit Test Test cases.....	45
7.4 Integration Testing.....	49
7.4.1 Integration Testing test cases.....	49

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT DESCRIPTION

Learn Sync is a web-based academic management system that offers a single digital platform for managing key academic processes in higher education institutions. The project aims to meet the academic tracking needs of postgraduate students, particularly in areas like internship submissions, project proposal evaluation, MOOC course records, faculty-student mentorship, and classroom content delivery (ClassOps). By combining the modules into a single platform, Learn Sync give students, faculty, and HOD' smoother workflows, and greater academic transparency.

The system has role-based authentication and dashboards. This allows students to upload academic data, such as internship certificates, project descriptions, and MOOC completions. Faculty can access student submission, evaluate and reject projects, guide mentees through mentorship records, and manage classroom announcements and submissions. The head of department (HOD) has administrative access for viewing and uploading bulk student and faculty data, track student progress, and ensure oversight within the institution.

Learn Sync platform brings automation and efficiency to an environment which is often weighed down by manual paperwork and slow academic tracking. This platform boosts student engagement, simplifies faculty tasks, and helps in making decisions based on data. The system's dynamic keyword-based project rejection analytics, mentorship insights, and user-friendly interface make it a valuable tool for the future of academic work.

## CHAPTER 2

# LITERATURE SURVEY

## 2.1 EXISTING AND PROPOSED SYSTEM

### 2.1.1 EXIXTING SYSTEM

In many educational institutions, managing student academic records, such as internships, project submissions, MOOC completions, and mentorship activities, is still a manual or semi-digital process. Typically, student submits physical documents or email them to faculty, who then review and file them manually. Mentorship detail and internal marks are often tracked using spreadsheets or paper files. This results in a lack of centralized oversight and delayed feedback. Classroom interaction like assignments submissions and announcements happen on multiple different platforms. This creates communication gaps and inconsistency in data.

### 2.1.2 PROPOSED SYSTEM

To solve these issues, the proposed Learn Sync system offer an unified web-based solution that get all major academic workflows into one digital platform. It provides a clear way for students to submit academic data and documents. Faculty can manage mentorship and classroom activities easily, and the HOD has complete administrative control based on their department. The system features automated project analysis before submission, secure document handling, real-time data tracking, and a login interface based on the user roles. Instead of separate tools, Learn Sync combines all features into one system, making it efficient and scalable.

## 2.2 FEASIBILITY STUDY

Before implementing Learn Sync, a feasibility study was conducted to determine whether the system would be viable, practical, and beneficial. The study was divided into the following areas:

- **Technical Feasibility:** The system uses proven technologies like HTML, CSS, JavaScript, PHP, and MySQL—all of which are widely supported, well-documented, and compatible with standard web hosting environments like XAMPP. The simplicity of the tech stack makes development, deployment, and maintenance cost-effective and technically manageable.

- **Operational Feasibility:** The system is user-friendly, requiring minimal training for students, faculty, and administrators. Since it mimics familiar workflows and interfaces (e.g., Google Classroom style for ClassOps), the learning curve is minimal. User roles are clearly defined, and dashboards are customized based on access levels, ensuring smooth operations across all modules.
- **Economic Feasibility:** As the project uses open-source tools and runs on local or cloud-hosted environments, the cost of development and deployment is low. No expensive licenses or proprietary software are required, making the system financially viable for educational institutions.
- **Schedule Feasibility:** The project was planned and completed within the academic timeline allotted for final-year MCA projects. Module-wise implementation and testing allowed for structured development and successful on-time delivery.

## 2.3 TOOLS AND TECHNOLOGIES USED

The following tools and technologies were used to develop and deploy Learn Sync:

- **Front-End:**
  - HTML5, CSS3 – For page layout and styling
  - JavaScript – For form validation and interactivity
  - Bootstrap – For responsive design and components
  - Tailwind CSS – For modern UI enhancements and dark mode support
- **Back-End:**
  - PHP – For server-side scripting and database interaction
  - MySQL – For data storage and retrieval
  - XAMPP – As a local development server environment
- **Development Tools:**
  - Visual Studio Code – Source code editor
  - phpMyAdmin – GUI tool for managing MySQL databases
  - Git – For version control
  - PHPSpreadsheet – For processing Excel uploads (student/faculty records).

## CHAPTER 3

# SOFTWARE REQUIREMENTS SPECIFICATION

### 3.1 INTRODUCTION

Learn Sync is a web-based academic management platform designed to integrate and streamline various academic activities within an educational institution. The system provides required modules for managing Internship, Project, MOOC courses, Mentorship, and a Classroom Operations Module (ClassOps). The application supports role-based access for Students, Faculty, and the HOD, each with dedicated dashboards and permission levels. This platform replaces manual and fragmented workflows with a centralised solution that enables students to submit academic data, faculty to review and guide students effectively, and the HOD to manage data and monitor the progress. The project includes a Project Evaluation & Approval system that analyzes project titles and descriptions using a keyword-based rejection engine, a Mentorship Module for tracking student growth and internal marks, and a ClassOps Module where faculty can post assignments, materials, and announcements for digital classrooms.

#### 3.1.1 PURPOSE

The purpose of Learn Sync platform is to build a centralised, digital platform that handles academic records and interaction between students and the faculty. It ensures seamless tracking of internship submissions, project evaluations, MOOC certifications, mentorship feedback, and class-level communications. This eliminates the inefficiencies of the paper-based record management and promotes transparency. The platform also automates project feasibility checks and supports bulk uploads, advanced search, secure file handling, and role-based data access.

#### 3.1.2 DOCUMENT CONVENTIONS / DEFINITIONS AND ABBREVIATIONS

- Learn Sync: The proposed academic management platform.
- HOD: Head of Department with administrative access.
- MOOC: Massive Open Online Courses.
- Mentorship Module: Allows students to input aspirations, hobbies, and internal marks.
- ClassOps: Classroom Operations module that enables assignment postings, content sharing, and digital submissions.

- PDF: Portable Document Format.
- LMS: Learning Management System.
- SIS: Student Information System.
- XAMPP: Cross-platform, Apache, MySQL, PHP, and Perl.
- PHP: Hypertext Preprocessor.
- HTML: Hypertext Markup Language.
- CSS: Cascading Style Sheet.
- LAN: Local Area Network.
- ER: Entity Relationship

### 3.1.3 INTENDED AUDIENCE

This Software Requirements Specification (SRS) document is intended for the following stakeholders who are directly involved in the design, development, testing, and evaluation of the Learn Sync system:

- **System Developers:** To understand the functional and non-functional requirements for implementing the platform.
- **Test Engineers:** To design appropriate test cases and ensure all modules meet specified requirements.
- **Project Guide and Academic Evaluators:** To review the technical scope, ensure completeness, and assess feasibility of the proposed system.
- **System Architects/Analysts:** To analyze system behavior, constraints, and dependencies for informed design decisions.
- **Documentation Reviewers:** To validate the structure, clarity, and accuracy of the specifications documented here.

### 3.1.4 PROJECT SCOPE

Learn Sync includes the following modules and functionalities:

- **Secure Login and Dashboards:** Role-based authentication system that provides personalized dashboards for students, faculty, and the HOD to access their respective modules securely.
- **Internship Module:** Allows students to submit internship details such as company name, role, duration, and upload supporting certificates in PDF or image formats.
- **Project Evaluation Module:** Enables students to check the feasibility of project titles using a backend keyword-based analysis engine and a historical rejection database.

- **MOOC Module:** Facilitates the submission and validation of completed online courses by capturing course name, platform, certification, and duration.
- **Mentorship Module:** Lets students submit personal details like hobbies, career goals, and semester-wise internal marks for faculty review and mentoring.
- **ClassOps Module:** Provides a digital classroom environment where faculty can post announcements, share resources, and collect student submissions systematically.
- **Bulk Data Upload:** HOD can upload structured Excel files to add multiple student and faculty records at once, reducing manual entry and improving accuracy.
- **Analytics and Reports:** Faculty and HODs can view filtered student records and generate printable academic reports to assist in evaluation and departmental review.

### 3.1.5 BENEFITS

- **Centralized Academic Management:** Learn Sync unifies various academic workflows—including internships, projects, MOOCs, mentorship, and classroom tasks—into a single, accessible digital platform.
- **Efficient Project Review:** The system offers a smart project evaluation engine combined with manual faculty approval, ensuring quality project selection and reducing duplication of ideas.
- **Mentorship Tracking:** Faculty can monitor students' academic progress, internal marks, hobbies, and career aspirations, allowing for personalized mentoring and timely academic support.
- **Classroom Collaboration:** Through the ClassOps module, faculty can manage digital classrooms by posting announcements, sharing materials, and collecting submissions from students seamlessly.
- **Secure and Scalable:** With role-based access control and a modular architecture, the system ensures data protection and can scale to accommodate growing student and institutional needs.

### 3.1.6 REFERENCES

- [1] Sommerville, I. (2015) – Software Engineering (10th Edition), Pearson.
- [2] Pressman, R. S. (2019) – Software Engineering: A Practitioner's Approach (9th Edition), McGraw-Hill.
- [3] MySQL Documentation (2023) – MySQL 8.0 Reference Manual, Retrieved from <https://dev.mysql.com/doc/>

[4] W3Schools (2023) – PHP and JavaScript Tutorials, Retrieved from

<https://www.w3schools.com/>

## 3.2 OVERALL DESCRIPTION

### 3.2.1 IDENTIFICATION OF PRE-EXISTING WORK

Before the development of Learn Sync, student academic data such as internship records, project proposals, MOOC course completions, and mentorship notes were maintained through a mix of physical files, email threads, spreadsheets, and loosely organized digital repositories. This method of managing the academic data led to inefficient workflows, frequent data losses, manual errors, and inconsistent reporting across the stakeholders. There was also no structured way to analyse project titles for duplication or track student growth continuously through mentorship insights. While the existing Learning Management Systems (LMS) and the Student Information Systems (SIS) addressed basic academic operations, they lacked customization capabilities to support department specific functions such as mentorship mapping or intelligent project analysis. These gaps highlighted the need for a comprehensive, integrated platform tailored to internal academic processes - thus leading to the creation of Learn Sync.

### 3.2.2 PRODUCT PERSPECTIVE

Learn Sync is an all-in-one web-based platform which caters specifically to the academic and administrative needs of the academic programs. It acts as a centralised system that replace disjointed tools and manual methods with automated workflows, improving visibility and control over the student academic records. The system adopt a modular architecture, which allows each module (Internship, Project, MOOC, Mentorship, ClassOps) to function both as independent and in coordination with others. This ensures scalability, easy maintenance, and future extensibility, should the institution wish to add more features. The platform is designed to work efficiently in an institutional environment and is aligned with the academic evaluation patterns of modern universities, making it a highly relevant solution.

### 3.2.3 PRODUCT FEATURES

- Integrated Platform: This provides a single interface where students, faculty, and the HOD can collaborate and manage all the academic submissions and reviews from one place, reducing duplication and delays.
- Automated Project Evaluation: Uses backend keyword analysis to assess new project titles, comparing them against a database of previously rejected topics to generate an

- acceptance/rejection probability.
- Mentorship Module: Empowers faculty to offer better guidance by accessing detailed student profiles, including personal interests, academic challenges, career goals, and semester-wise internal marks.
  - Digital Classrooms (ClassOps): Mimics features of modern classroom management tools by allowing faculty to create online classrooms, share announcements and materials, and manage student submissions digitally.
  - Role-Based Access Control: Ensures secure and segregated access based on user roles, maintaining data privacy and enabling customized dashboards tailored to each user's responsibilities and permissions.
  - Data Upload and Reporting: Provides the HOD with options for bulk uploading student and faculty data using Excel files, along with tools for generating structured reports for academic audits and performance reviews.

### 3.2.4 END USER CHARACTERISTICS

- **Students:** Submit details related to internships, projects, MOOCs, mentorship, and participate in ClassOps activities. Users are expected to have basic computer and internet navigation skills.
- **Faculty:** Evaluate student submissions, approve/reject projects, view mentorship records, and manage classroom materials and assignments.
- **HOD (Administrator):** Oversees system-wide data, uploads student/faculty information in bulk, assigns faculty mentors, and monitors academic progress.

### 3.2.5 OPERATING ENVIRONMENT

- Platform – Web-based application accessible via modern browsers (Google Chrome, Mozilla Firefox, Microsoft Edge).
- Backend – PHP, MySQL (Database for storing records).
- Frontend – HTML, CSS, JavaScript (Ensuring a user-friendly interface).
- Server – Hosted on Apache using XAMPP for local development.
- Device Compatibility – Responsive design for desktops, laptops, and mobile devices.

### 3.2.6 DESIGN AND IMPLEMENTATION CONSTRAINTS

- **Security & Privacy:** Must ensure proper access restrictions and secure handling of academic documents and user credentials.
- **Keyword Processing Efficiency:** The backend for project evaluation must handle keyword extraction and comparison efficiently for timely suggestions.
- **Training Requirements:** Initial orientation may be needed for students and faculty unfamiliar with academic web portals.
- **Internet Dependency:** Requires stable internet connectivity for data uploads, real-time access, and class operations.

### 3.2.7 ASSUMPTIONS AND DEPENDENCIES

- **Users Have Basic Technical Knowledge:** It is assumed that all users of the system, including students, faculty, and HODs, have a basic level of computer skills and know how to use web-based platforms. This includes navigating user dashboards, submitting forms, uploading files, and understanding on-screen instructions with little training or technical help.
- **Availability of Internet & Server Infrastructure:** The smooth operation of Learn Sync depends on a stable internet connection and a properly set up server environment, such as XAMPP or something similar. Any interruptions in the network or server may limit access to important features like data uploads, certificate viewing, or classroom interactions.
- **Regular Database Maintenance and Backup:** To keep data safe and the system stable, it is assumed that the hosting environment will perform regular database backups and scheduled maintenance tasks. This will help avoid data loss from hardware failures, accidental deletions, or unexpected technical problems.
- **System Compatibility with Updated Web Browsers:** The system assumes that users will access Learn Sync through modern, updated versions of standard web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari. Full functionality, including interactive features and layout rendering, depends on these browsers being compatible with the underlying front-end technologies.

### 3.3 PRODUCT FUNCTIONALITY

Learn Sync is a role-based academic management system where each user type—Admin (HOD), Faculty, and Student—has access to distinct functionalities based on their responsibilities within the academic ecosystem. The system ensures secure data handling, seamless form submissions, and real-time collaboration between users through module-specific operations.

#### 3.3.1 MODULE DETAILS

##### 1. Admin Module (HOD)

The Admin or Head of Department has full access to the system's data, allowing them to manage users, monitor academic records, and oversee system-wide operations. Key functionalities include:

- **User Management:** Upload and manage student and faculty details in bulk using Excel files, reducing manual entry and potential errors.
- **Record Monitoring:** View complete academic records of all students, including internship details, project submissions, MOOC completions, mentorship forms, and ClassOps participation.
- **Faculty Assignment:** Assign faculty members to students for mentorship and project evaluation, enabling structured student guidance.
- **Report Generation:** Generate detailed academic performance reports filtered by semester, module, or department for administrative review and evaluation.
- **Classroom Oversight:** Monitor classroom activities within ClassOps, including posted content and student submissions.

##### 2. Faculty Module

Faculty members act as evaluators and mentors for the students assigned to them. They are responsible for reviewing academic progress, providing feedback, and facilitating classroom interactions. Key functionalities include:

- **Review Student Records:** Access and validate internship data, project details, and MOOC course completions submitted by students.
- **Project Evaluation & Approval:** View the project acceptance/rejection percentage calculated by the backend engine and make final decisions to approve or reject student proposals.

- **Mentorship Tracking:** Access student-submitted mentorship forms containing personal interests, career aspirations, and internal marks, helping provide tailored academic guidance.
- **ClassOps Management:** Create digital classrooms, post announcements and assignments, upload study materials, and manage student submissions in an organized digital space.
- **Performance Monitoring:** Generate and download academic performance reports of students under their guidance, supporting data-driven mentorship.

### 3. Student Module

Students are the primary users responsible for submitting their academic information and engaging in mentorship and classroom-related activities. Key functionalities include:

- **Internship Submission:** Submit internship-related details such as company name, role, duration, location, and upload supporting certificates for verification.
- **Project Proposal Submission:** Enter project titles and descriptions to receive automated rejection/acceptance analysis. Modify and resubmit titles based on faculty feedback if needed.
- **MOOC Course Upload:** Add details of completed online courses including course name, platform, duration, and certificate uploads.
- **Mentorship Form Submission:** Fill out comprehensive mentorship forms detailing hobbies, career goals, and semester-wise internal marks for academic tracking.
- **ClassOps Participation:** Join digital classrooms created by faculty, view materials, submit assignments, and receive updates and announcements.
- **Status Tracking:** Monitor the review status of submitted data such as project approvals, internship certificates, and faculty feedback through the student dashboard.

## 3.4 EXTERNAL INTERFACE REQUIREMENTS

### 3.4.1 USER INTERFACES

The Learn Sync platform provides a clean, responsive, and modular user interface tailored to each user type—Student, Faculty, and HOD. The UI is built using HTML5, CSS3, JavaScript, and Bootstrap/Tailwind CSS, ensuring accessibility and mobile responsiveness.

- **Student Dashboard:** A student-friendly panel showing options to submit internship, project, MOOC, and mentorship data, view statuses, upload certificates, and access ClassOps materials.

- **Faculty Dashboard:** Allows faculty to review student submissions, approve or reject project titles, view mentorship forms, manage ClassOps content, and generate performance reports.
- **HOD Dashboard:** Provides administrative tools to bulk upload student/faculty data, assign mentors, access full student records, and print/export academic reports.

Each interface uses validation prompts, file upload components, navigation bars, and modular forms for an intuitive user experience.

### 3.4.2 HARDWARE INTERFACES

Learn Sync is a web-based application and does not require any dedicated or proprietary hardware interface. However, for smooth operation, the following basic hardware specifications are assumed:

- **Client-Side (Students/Faculty/HOD):**
  - Device: Desktop, laptop, tablet, or smartphone
  - RAM: Minimum 2 GB
  - Processor: Dual-core 2.0 GHz or higher
  - Display: 1024×768 resolution or higher
  - Input: Standard keyboard and mouse or touch screen
- **Server-Side (XAMPP Hosting):**
  - Processor: Quad-core or higher
  - RAM: Minimum 4 GB
  - Storage: Minimum 10 GB (expandable as per records)
  - Connectivity: Stable internet connection for deployment

### 3.4.3 SOFTWARE INTERFACES

The system interacts with several open-source and standard software components for development, hosting, and database management:

- **Frontend:**
  - HTML5, CSS3, JavaScript
  - Bootstrap/Tailwind CSS for UI styling
- **Backend:**
  - PHP (Server-side scripting)
  - MySQL (Relational database)

- **Server Stack:**
  - Apache Server via XAMPP
  - phpMyAdmin for database interface
- **Other Tools:**
  - Git (Version control)
  - PHPSpreadsheet (for Excel file handling)

These software components work in sync to enable seamless operation of all modules within Learn Sync.

#### 3.4.4 COMMUNICATION INTERFACES

Since Learn Sync is a web-based platform, HTTP/HTTPS protocols are used for all communication between client and server. All interactions happen over a local or cloud-hosted server environment through standard web technologies.

- **Browser Communication:** The system uses asynchronous requests (AJAX) for dynamic form submissions and status updates without reloading the page.
- **File Uploads and Downloads:** Users can upload documents such as internship certificates, project reports, and MOOC proofs using form-based file upload mechanisms. Faculty can download reports or student documents directly via the dashboard.
- **Internal Navigation and Routing:** Navigation is handled through server-side routing via PHP and internal linking between user modules, ensuring a smooth and responsive browsing experience.

In future enhancements, optional integrations such as email notifications or SMS alerts could be added for improved communication.

### 3.5 NON-FUNCTIONAL REQUIREMENTS

#### 3.5.1 PERFORMANCE REQUIREMENTS

- The system is designed to handle simultaneous logins from multiple users, including students, faculty, and the HOD, without performance degradation.
- Typical operations such as data submissions, uploads, and dashboard loads are optimized to respond within 2–3 seconds under average load.
- The backend logic for project keyword analysis is built to return acceptance/rejection percentages in real-time (under 1 second).

- Batch uploads (via Excel files) are processed within a few seconds depending on file size and server specifications.

### 3.5.2 SAFETY REQUIREMENTS

- The system implements secure login protocols to prevent unauthorized access.
- All uploaded files (certificates, reports) are stored in protected directories on the server to avoid tampering or unauthorized downloads.
- User sessions are terminated after inactivity to reduce risk of misuse.
- Regular database backups must be scheduled to ensure data recovery in case of system failure or corruption.

### 3.5.3 SOFTWARE QUALITY ATTRIBUTES

- **Usability:** The system is designed with a user-friendly, clean, and intuitive interface, ensuring that students, faculty, and HODs can navigate it with minimal learning curve. Clear labeling, structured dashboards, form validation messages, and responsive layouts improve user satisfaction and reduce the chances of user errors during data entry or uploads.
- **Maintainability:** Learn Sync is built with a modular and organized codebase, making it easy to identify, isolate, and fix issues without affecting other modules. The backend is structured with reusable components, allowing developers to perform upgrades, enhancements, or bug fixes efficiently as the system scales or evolves over time.
- **Portability:** As a browser-based application, Learn Sync is platform-independent and can be accessed from a wide range of devices including desktops, laptops, tablets, and smartphones. It does not require any installation and functions well across various operating systems like Windows, macOS, and Linux, provided a modern browser is used.
- **Reliability:** The system has been tested to handle concurrent users and multiple operations without data loss or performance breakdown. Built-in validation, file upload restrictions, and proper session handling ensure that the system behaves consistently under normal institutional loads, thereby establishing trust among users.
- **Security:** Robust role-based access control (RBAC) ensures that only authorized users can access or modify data relevant to their role. Student data, academic submissions, and certificate uploads are protected using server-side validation and restricted directory access. Additionally, logout timers and secure data handling practices reduce the risk of unauthorized access or data breaches.

### 3.5.4 SPECIFIC REQUIREMENTS

- Excel file uploads must adhere to a predefined format to ensure correct mapping of student and faculty data into the database.
- File size limits for certificate uploads (PDF, PNG, JPG) should be enforced to prevent server overload.
- Input validation is mandatory across all modules to avoid null entries or SQL injection vulnerabilities.
- The system must support certificate preview for PDF and image files using an embedded viewer.

## 3.6 OPERATING ENVIRONMENT

The Learn Sync system is developed and deployed in a controlled web-based environment, designed to support the academic and administrative needs of higher education institutions. It operates effectively on a local or institutional server using open-source technologies and requires a standard client-server setup with browser access.

The system is tested and optimized for use within LAN-based college networks or internet-enabled environments. It assumes users will access the system through updated browsers and standard computing devices.

### 3.6.1 HARDWARE

#### Client-Side Requirements (Students, Faculty, HOD):

- **Processor:** Dual-core 2.0 GHz or higher
- **RAM:** 2 GB minimum
- **Display:** 1024×768 resolution or higher
- **Storage:** Sufficient for temporary browser cache
- **Devices Supported:** Desktop, Laptop, Tablet, Smartphone
- **Input Devices:** Standard keyboard, mouse, or touchscreen

#### Server-Side Requirements:

- **Processor:** Quad-core or higher
- **RAM:** Minimum 4 GB (8 GB recommended for scalability)
- **Storage:** 20 GB or more (based on expected certificate and data uploads)
- **Network:** Stable internet or LAN connection for intranet hosting
- **Backup Device:** External storage or cloud backup setup for regular database dumps

### 3.6.2 SOFTWARE

#### Client-Side:

- **Operating System:** Windows 10/11, Linux, macOS, Android (browser access)
- **Browsers Supported:** Chrome, Firefox, Microsoft Edge, Safari (latest versions)

#### Server-Side:

- **Operating System:** Windows/Linux (for XAMPP deployment)

- **Stack:**

- Apache (via XAMPP)
- PHP 7.x or higher
- MySQL 5.7 or higher

- **Tools Used:**

- phpMyAdmin (for database interface)
- Visual Studio Code (for development)
- Git (for version control)
- PHPSpreadsheet (for Excel import/export)

The system is fully compatible with the XAMPP stack and can be deployed on both local machines and hosted cloud servers if needed.

### 3.7 DELIVERY PLAN

The development of Learn Sync was structured into distinct phases aligned with the academic semester timeline. Each phase contributed to a progressive, modular build ensuring functionality, usability, and timely completion of the system.

The project followed a milestone-based approach where each module was independently developed and tested before integration. This ensured smooth transitions between stages and minimized delays during the final deployment and documentation process.

#### Phase 1 - Requirement Gathering and Analysis

In this phase, the foundational structure of the project was established. Stakeholders' requirements were discussed and documented to determine the module list, user roles, and workflows.

##### **Duration:** Week 1

- Identified academic workflows and defined system requirements.
- Outlined key modules: Internship, Project, MOOC, Mentorship, and ClassOps.

- Designed the ER diagram and database schema.

## Phase 2 - Backend & Database Development

This phase focused on setting up the server-side logic and building a normalized, scalable database schema. Backend scripts were developed to support data submissions and user authentication.

**Duration:** Weeks 2–3

- Created MySQL tables for all modules.
- Developed PHP scripts for form handling and database interactions.
- Implemented Excel-based bulk data upload for student and faculty records.

## Phase 3 - Frontend UI Design and Integration

The system's interface was built to provide role-based dashboards and forms, ensuring an intuitive experience for all users. Frontend components were then connected to backend logic for seamless operation of the system.

**Duration:** Weeks 4–5

- Designed dashboards and forms using HTML, CSS, Bootstrap, and Tailwind CSS.
- Integrated frontend with backend logic and added JavaScript validation.
- Ensured responsive design for desktops and mobile devices.

## Phase 4 - Testing and Debugging

All modules underwent extensive testing to identify and resolve bugs and usability issues. Emphasis was placed on ensuring the reliability, performance, and accuracy of the system.

**Duration:** Week 6

- Conducted form validation, file upload testing, and login tests.
- Performed role-based functionality checks for students, faculty, and HOD.
- Fixed bugs and optimized performance for real-time data processing.

## Phase 5 - Final Deployment and Documentation

This final stage involved system deployment and preparation of academic deliverables. The completed project was demonstrated successfully with all modules functional.

**Duration:** Week 7

- Deployed the system on XAMPP (local server) for demonstration.
- Prepared project documentation: Synopsis, SRS, and Final Report.
- Successfully presented Learn Sync with all modules functioning as intended.

## CHAPTER 4

# SYSTEM DESIGN

### 4.1 INTRODUCTION

System design is a critical phase in the software development life cycle, serving as the bridge between user requirements and system implementation. It involves defining the architecture, components, modules, data flow, and interactions within the system to ensure it functions as intended. For the LEARN SYNC project, system design focuses on identifying various user roles such as students, faculty, and HODs—and outlining how their functionalities are structured, accessed, and executed within the platform. This stage translates functional requirements into detailed specifications that guide the actual development. It includes both logical design, which defines the flow of data and processing logic, and physical design, which involves the technical structure such as databases, interfaces, and communication layers. A well-structured system design ensures that all components of LEARN SYNCare seamlessly integrated, scalable, secure, and capable of meeting the academic management needs of the entire campus.

### 4.2 SCOPE OF THE PROJECT

The Learn Sync project is designed to serve as a centralized, role-based academic management platform that streamlines the coordination of curricular and co-curricular activities across departments such as Engineering, MBA, and MCA. The system fosters collaboration among students, faculty, and HODs while ensuring secure data handling, seamless integration of academic records, and partial automation of department-level processes. The major areas within the scope of this project include:

- **User Authentication and Role-Based Access:** A secure login system with differentiated dashboards and functionalities for Students, Faculty, and HODs, ensuring that users access only their authorized modules.
- **Internship, Project, and MOOC Management:** Enables students to submit details of their internships, academic projects, and MOOC completions, along with the facility to upload corresponding certificates for review.
- **Mentorship Module:** Facilitates entry of student-specific academic, personal, and career-related data, which can be accessed and evaluated by faculty mentors to guide student development.

- **Project Evaluation and Approval System:** Provides automated analysis of project titles using keyword-based logic, displaying approval/rejection probabilities and allowing faculty to override suggestions based on academic discretion.
- **ClassOps Module (Classroom Operations):** Enables faculty to create digital classrooms, post materials, assignments, and announcements, while students can view and submit responses—similar in function to Google Classroom.
- **Bulk Data Upload by HOD:** HODs can upload Excel files containing structured student and faculty data, which are validated and inserted into the database automatically to reduce manual work.
- **Dynamic Record Viewing and Report Generation:** Faculty can access and review records of assigned students, whereas HODs can view all departmental data and generate printable reports for administrative use.
- **Secure File Management System:** All certificate uploads, including those for internships, projects, and MOOCs, are securely stored in the server and linked to each student's academic record.
- **Search and Filter Capabilities:** Integrated search bars and dropdown filters allow users to quickly locate specific student records based on criteria such as name, USN, or course type.
- **Responsive and Modular Design:** The platform is mobile-friendly, built with scalability in mind, and organized into independent modules to simplify future upgrades or institutional expansion.

## 4.3 INTENDED AUDIENCE

The Learn Sync platform is designed for use by a broad range of stakeholders within a higher education institution, particularly in departments managing student academic records, co-curricular activities, and faculty-student interaction. Each user group interacts with the system differently based on their roles, responsibilities, and access levels. The system has been tailored to meet the unique functional needs of the following audiences:

### 4.3.1 Students

- Students form the primary user base of the system and are provided with intuitive interfaces to manage and track their academic engagements.
- Upload and manage internship details, project proposals, MOOC certifications, and mentorship-related inputs.
- Monitor the status of project evaluations including automated analysis and faculty

decisions.

- Participate in digital classrooms via the ClassOps module for accessing study materials and submitting assignments.
- Update personal details and securely change account passwords when needed.

#### **4.3.2 Faculty Members**

- Faculty members use the system to manage student mentorship, evaluate academic submissions, and conduct class operations.
- View academic, personal, and mentorship records of students assigned to them.
- Evaluate and optionally reject project titles, even if accepted by the backend title analysis algorithm.
- Manage digital classrooms by uploading content, posting announcements, and receiving student submissions through ClassOps.
- Track participation in MOOCs, internships, and mentorship activities for individual mentees.

#### **4.3.3 Head of Department (HOD)**

- The HOD acts as the department administrator and oversees the entire data flow across all modules.
- Perform initial bulk uploads of student and faculty data using Excel templates.
- Access academic records across all modules for every student within the department.
- Generate printable reports and insights to assist in departmental audits, academic reviews, and compliance tracking.

#### **4.3.4 Developers and System Maintainers**

- This group includes technical stakeholders responsible for deploying, maintaining, and extending the system.
- Understand and work with the system's architecture, code structure, and database schema for updates and debugging.
- Monitor performance, apply security patches, and implement future enhancements based on user feedback or institutional needs.
- Ensure system reliability and scalability across multiple departments and academic years.

#### **4.3.5 Institutional Administrators and Evaluators**

- These include higher-level stakeholders such as accreditation bodies, quality assurance teams, and academic auditors.

- Utilize centralized access to department-level academic records for evaluation and benchmarking.
- Leverage printable reports for use during inspections, audits, NAAC/NBA accreditations, and internal quality assurance reviews.

## 4.4 SOFTWARE PRODUCT ARCHITECTURE

The architecture of the Learn Sync system as shown in fig 4.4.1 is based on a three-tier client-server model, ensuring modularity, scalability, and ease of maintenance. It separates the application logic, user interface, and data access into different layers, allowing each layer to be managed and updated independently without affecting the rest of the system.

This architectural model supports the system's multi-role functionalities - providing customized interfaces and controls for students, faculty, and the Head of Department (HOD). Each user interaction triggers processes that flow through the presentation layer to the business logic and then to the data layer for persistent storage.

### 4.4.1 ARCHITECTURAL DESIGN

The system adopts a modular web-based architecture using open-source technologies like PHP, MySQL, HTML/CSS, and JavaScript. The architecture ensures:

- Clear separation of concerns.
- Reusability of components.
- Easy integration of additional modules like mentorship and ClassOps.
- Secure and efficient communication between layers.

All major functionalities - project evaluation, mentorship, ClassOps, internship, and MOOC submissions - are implemented as separate modules working together within the same system framework.

#### 4.4.1.1 View Layer (Presentation Layer)

This layer is responsible for the user interface (UI) and overall user experience. Built using HTML, CSS (Bootstrap and Tailwind CSS), and JavaScript, it allows users to interact with the system through forms, dashboards, buttons, filters, and tables.

Role-specific dashboards are provided:

- Students view submission forms and progress status.
- Faculty access records and mentorship data.
- HOD can view, upload, and generate reports.

Responsiveness and mobile accessibility are key features of this layer.

#### 4.4.1.2 Business Layer

This layer contains the core logic of the application and handles all major processes.

Implemented using PHP, it manages:

- User authentication and session handling.
- Project title analysis and rejection logic.
- File uploads and validations.
- Certificate path storage and reporting logic.

It ensures that only validated and authorized requests are passed to the data layer.

#### 4.4.1.3 Access Layer (Data Layer)

The data layer is responsible for storing, retrieving, and updating data in the MySQL database.

It interacts with multiple tables such as students, faculty, internship, project, mooc\_courses, mentor\_form, classops\_posts, and internal\_marks.

PHP scripts are used to:

- Perform CRUD operations.
- Validate Excel uploads.
- Link uploaded files (certificates/documents) with student records.
- Ensure referential integrity through foreign keys and structured schemas.

All interactions with the database are secure, and queries are optimized to handle large volumes of data.

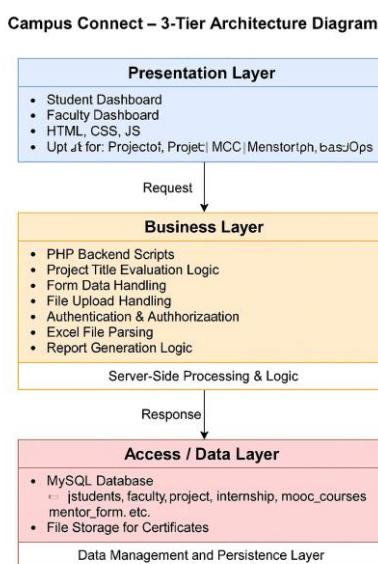


Fig 4.4.1: 3-Tier Architecture

This diagram illustrates the layered architecture of the system, dividing it into Presentation

(UI), Business Logic, and Data Access layers. It promotes modularity, separation of concerns, and makes the system easier to maintain and scale.

## 4.5 COMPONENT ARCHITECTURE

The Learn Sync system is structured into a modular component-based architecture. Each functional module is developed as an independent, reusable component, interacting seamlessly with other modules through a shared backend and unified database. This architecture enhances maintainability, simplifies debugging, and supports the easy addition of future modules.

The key components of the system are:

**a) Authentication & Role Management**

- Handles secure login and session management.
- Verifies user credentials and redirects to appropriate dashboards.
- Role-based access ensures users interact only with authorized components.

**b) Internship Management Module**

- Enables students to submit internship details along with certificate uploads.
- Faculty and HOD can view records and validate entries.
- Data is stored in the internship table with certificate path linking.

**c) Project Evaluation Module**

- Accepts project titles and performs backend keyword analysis.
- Calculates acceptance/rejection percentage based on past submissions.
- Faculty can override the decision by manually rejecting or accepting.
- Status and reason are stored in the project table.

**d) MOOC Course Module**

- Students can enter MOOC course completions.
- Certificates are uploaded for each course.
- Faculty and HOD can review and generate reports from the mooc\_courses table.

**e) Mentorship Module**

- Students submit personal info, aspirations, achievements, and internal marks.
- Faculty mentors can view and guide based on these records.
- Data is distributed across mentor\_form, student\_hobbies, student\_achievements, and internal\_marks tables.

**f) ClassOps (Classroom Operations)**

- Faculty can create digital classrooms, post announcements, and upload materials or assignments.

- Students can view posts and upload assignment submissions.
- Data managed using classops\_posts and classops\_submissions tables.

### g) HOD Administrative Tools

- Allows bulk upload of student and faculty data through Excel files.
- Enables record filtering, printing, and department-level progress tracking.
- Auto-validation during uploads ensures database consistency.

## 4.5.1 USER INTERFACE

The Learn Sync system provides clean, intuitive, and role-based user interfaces that are responsive and consistent across devices. Built using HTML, CSS (Bootstrap and Tailwind CSS), and JavaScript, the UI ensures a user-friendly experience for all stakeholders - students, faculty, and HODs.

### a) Student Dashboard

- View and submit details for internships, projects, MOOCs, and mentorship.
- Upload certificates and monitor project evaluation feedback.
- Participate in ClassOps (access materials, announcements, and assignments).
- Change password and update profile.

### b) Faculty Dashboard

- View assigned students and their submissions.
- Evaluate projects (approve or reject).
- Post materials and assignments in ClassOps.
- Monitor student progress in mentorship and academic activities.

### c) HOD Dashboard

- Upload Excel files for bulk student/faculty data insertion.
- View and print all student academic records.
- Access analytics and generate department-level reports.

The UI design follows a modular layout, separating functionalities into sections for ease of navigation. Dropdowns, search bars, form validation messages, and table views enhance usability and accessibility.

## 4.6 DATA FLOW DIAGRAM

The Data Flow Diagram (DFD) describes how data moves within the system from input to processing to output. It is represented in multiple levels for clarity.

#### 4.6.1 Level 0 - Context Level DFD

Depicts the overall system interacting with three external entities:

- **Students** submit data (project, internship, MOOC, mentorship).
- **Faculty** evaluate and review data.
- **HOD** manages administrative data and reporting.

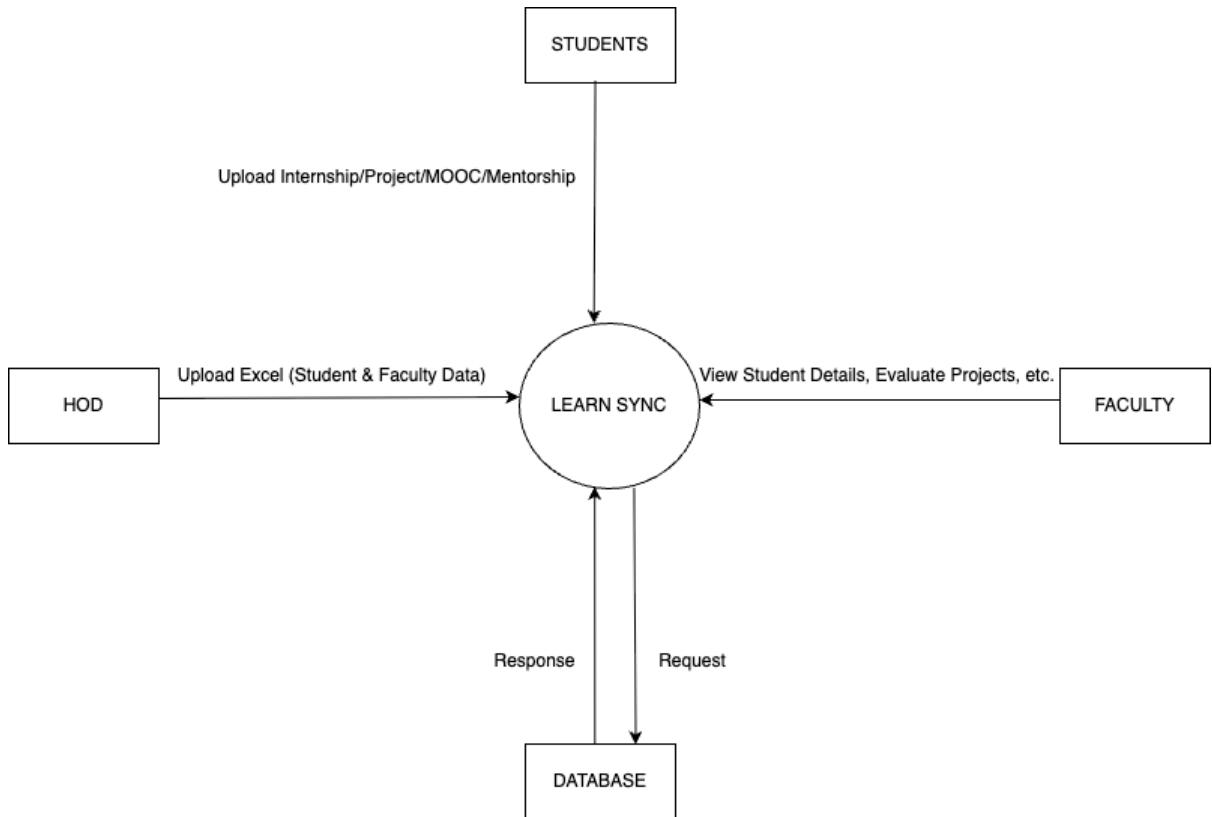


Fig 4.6.1 Level 0 DFD

This context-level DFD provides an overview of the entire Learn Sync system as a single process interacting with external entities such as Students, Faculty, and HOD. It outlines basic data exchange without internal module breakdown.

#### 4.6.2 Level 1 - Major Module Interactions

Shows internal processing:

- Student submits project title → sent to project analyzer → acceptance % → stored in DB.
- Faculty evaluates → accepts/rejects → status updated.
- Certificates uploaded → validated → stored securely.
- HOD uploads Excel files → parsed → inserted into DB.

#### Data Stores

- students, faculty, internship, project, mooc\_courses, mentor\_form, classops\_posts, etc.
- File storage: PDF/JPG/PNG certificates stored in structured directories.

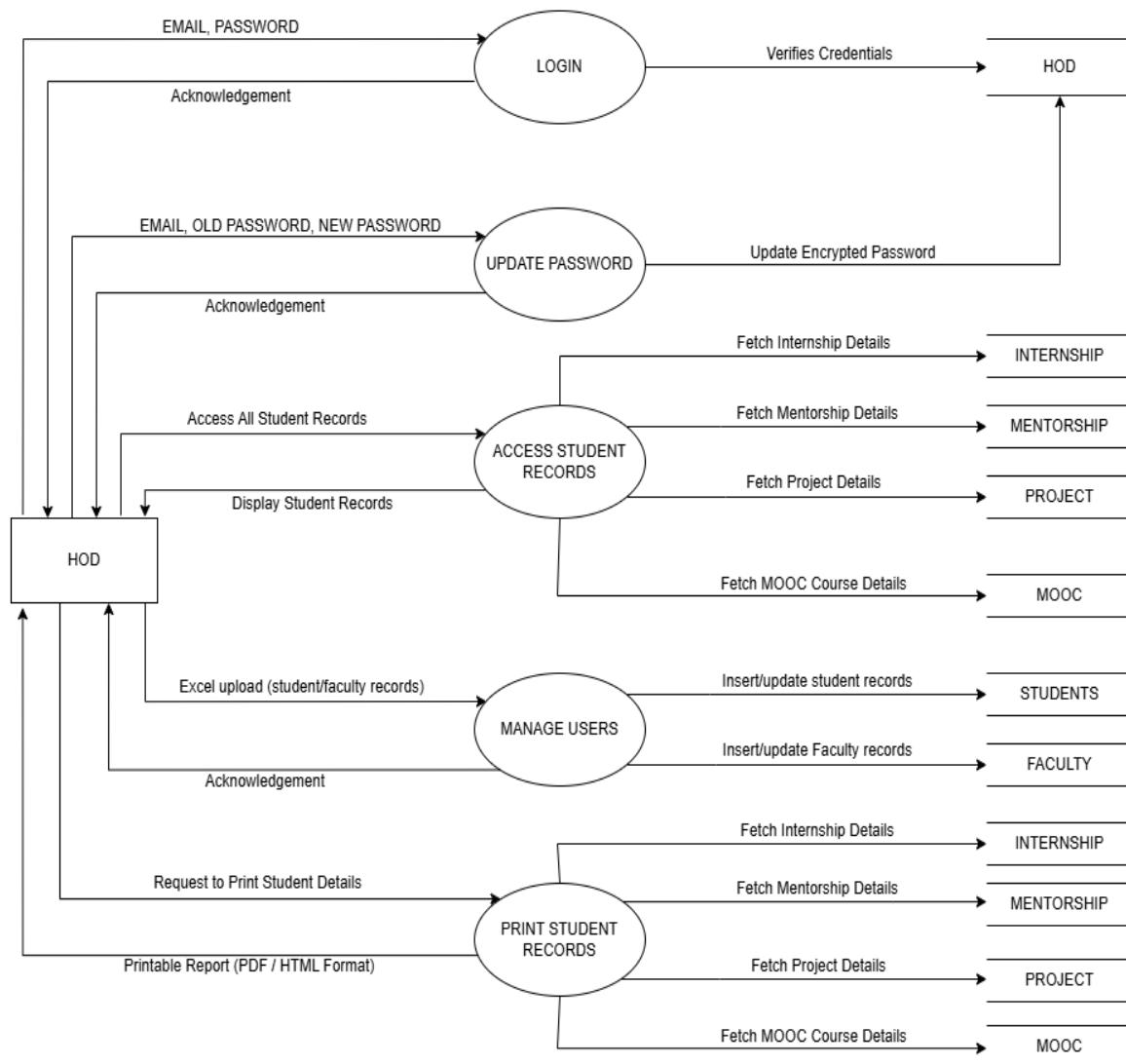


Fig 4.6.1 Level 1 DFD for HOD Module

This diagram depicts the administrative processes handled by the HOD, including uploading Excel files, managing user data, and viewing department-level analytics. It outlines HOD's control over data validation and access.

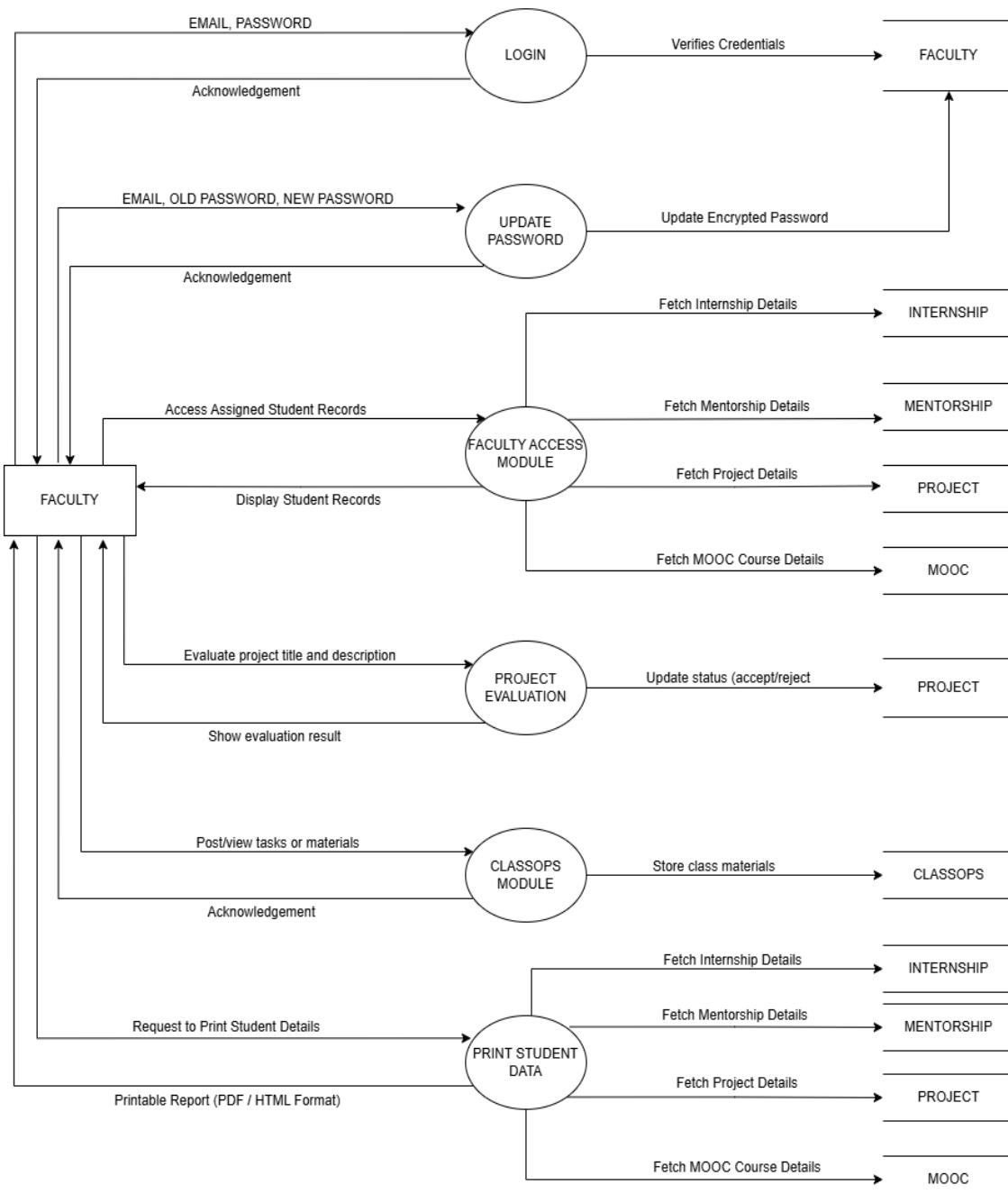


Fig 4.6.2 Level 1 DFD for Faculty Module

This DFD illustrates the flow of data for faculty members. It shows how faculty access student submissions, manage ClassOps materials, and interact with the database to review academic records.

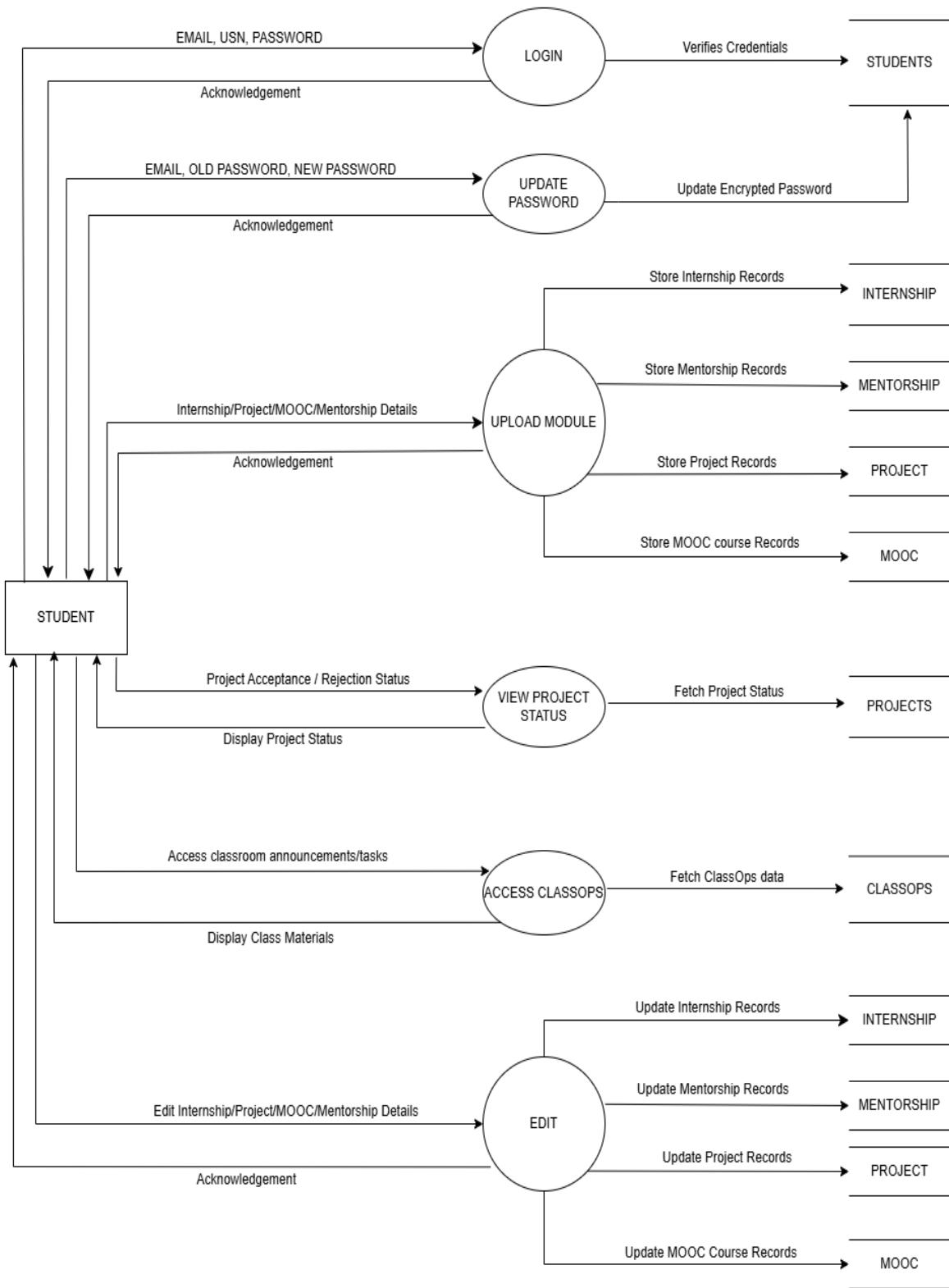


Fig 4.6.3 Level 1 DFD for Student Module

This Level 1 DFD focuses on the student's interaction within the system. It details the processes involved in submitting internship, project, MOOC, and mentorship data, and how the system stores and responds to this data.

## CHAPTER 5

### DETAILED DESIGN

#### 5.1 USE CASE DIAGRAM

The use case diagram visually represents the interaction between users (actors) and the system. In Learn Sync, the primary actors are Students, Faculty, and HOD. Each user type interacts with different modules depending on their access level as shown in fig 5.1.1.

##### **Student Use Cases:**

- Login securely
- Submit internship, project, and MOOC details
- Upload certificates
- View project rejection/acceptance status
- Participate in ClassOps (view posts, submit work)
- Fill mentorship form

##### **Faculty Use Cases:**

- Login securely
- View student records
- Evaluate/reject project titles
- Monitor MOOC and mentorship records
- Post content in ClassOps

##### **HOD Use Cases:**

- Login securely
- Upload Excel files (students/faculty)
- View, print, and filter all student data
- Generate reports

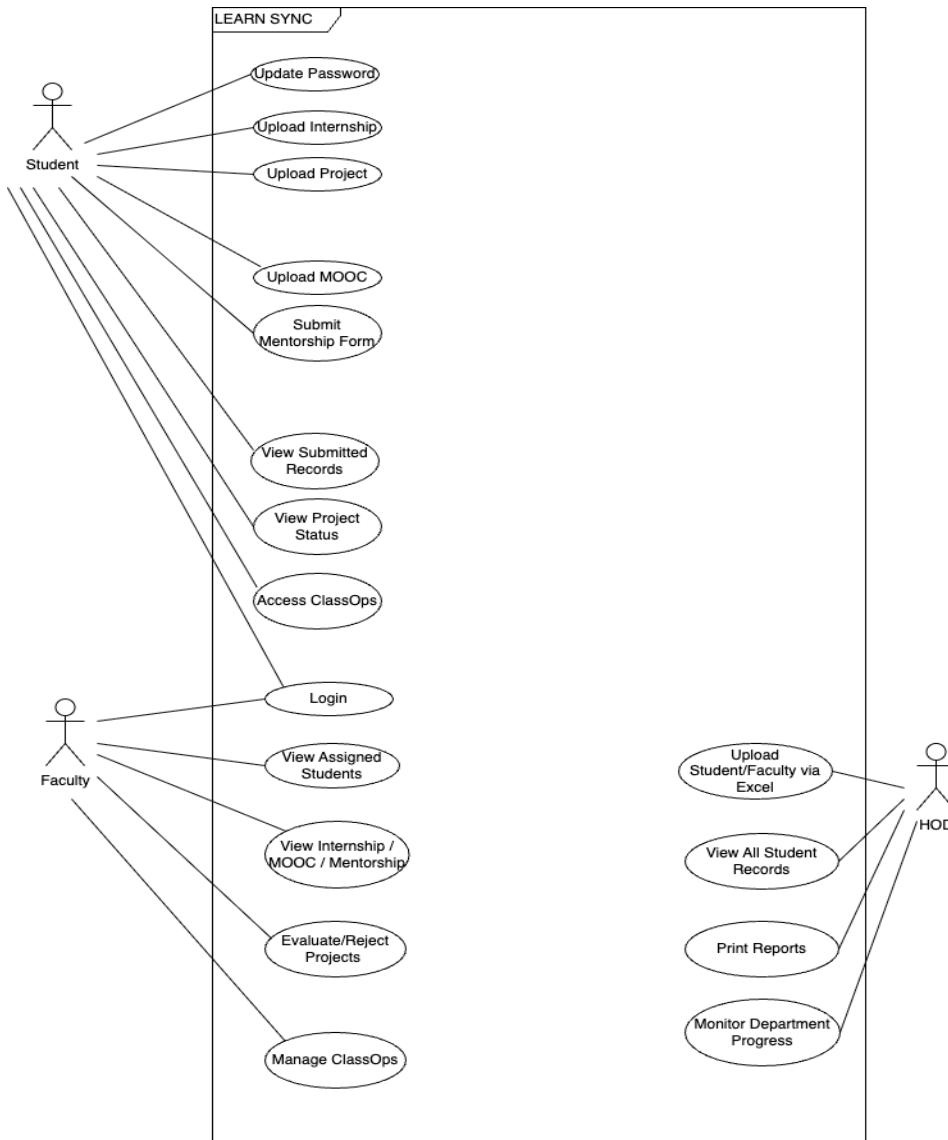


Fig 5.1.1 Use Case Diagram

The diagram illustrates the modular flow of responsibilities and highlights the role-based interactions, ensuring clear boundaries and security between functionalities.

## 5.2 SEQUENCE DIAGRAMS

Sequence diagrams define the chronological order in which components of the system interact. Each diagram provides a clear depiction of how Learn Sync manages requests and responses between users (HOD, Faculty, Student) and system layers (Server, Database, UI Modules).

### a) HOD Workflow Sequence:

This diagram as shown in fig. 5.2.1 describes the series of interactions that the Head of Department (HOD) performs for uploading data, monitoring progress, and printing reports.

**Flow:**

- HOD enters login credentials → system verifies via server and database.
- Once authenticated, HOD accesses the dashboard and selects the "Upload Student Excel" option.
- Upload panel is triggered and allows HOD to submit the Excel file.
- Server validates file, inserts records into the students table.
- A similar flow occurs for uploading the Faculty Excel file → records stored in faculty table.
- HOD selects "View Records" → system retrieves all student and faculty records from DB.
- When "Print" is selected, system generates a downloadable printable layout (PDF).
- HOD can also request performance overview reports → system aggregates and returns statistics.

This sequence ensures streamlined data onboarding and analytics access for department administrators.

#### **b) Faculty Dashboard Interaction Sequence:**

This diagram outlines how faculty interact with the system to evaluate students' academic work, review mentorship details, and manage ClassOps. Figure 5.2.2 shows the workflow sequence for faculty.

##### **Flow:**

- Faculty logs in → credentials validated → redirected to Faculty Dashboard.
- System fetches data of assigned students including MOOC and Internship entries.
- Faculty selects a project to evaluate → system displays project description + acceptance percentage.
- Faculty can override system's suggestion and choose to approve/reject → system updates DB.
- Faculty accesses mentorship forms → retrieves academic and personal data submitted by students.
- Through ClassOps, faculty creates posts or assignments → saved in classops\_posts.
- Students are notified about new updates or assignments.

This flow promotes mentor-student engagement and academic monitoring by simplifying data retrieval and feedback mechanisms.

#### **c) Student Interaction Sequence:**

This sequence captures all major actions performed by students—from login to submitting various forms and participating in classroom activities.

**Flow:**

- Student logs in → credentials validated → redirected to Dashboard.
- For each module:
  - **Internship:** Student enters details → uploads certificate → data stored in internship table.
  - **Project:** Student enters title → system checks for acceptance % → if submitted, stored in project table.
  - **MOOC:** Student fills form → uploads details → stored in mooc\_courses.
  - **Mentorship:** Student submits academic info, hobbies, aspirations → stored across multiple tables (mentor\_form, student\_hobbies, etc.).
- Student can view status of all submissions.
- Password updates are processed securely through the system.
- Student accesses ClassOps for materials and assignments → can submit work which is saved under classops\_submissions.

This sequence shows a complete student journey through the system, emphasizing usability, modularity, and academic engagement.

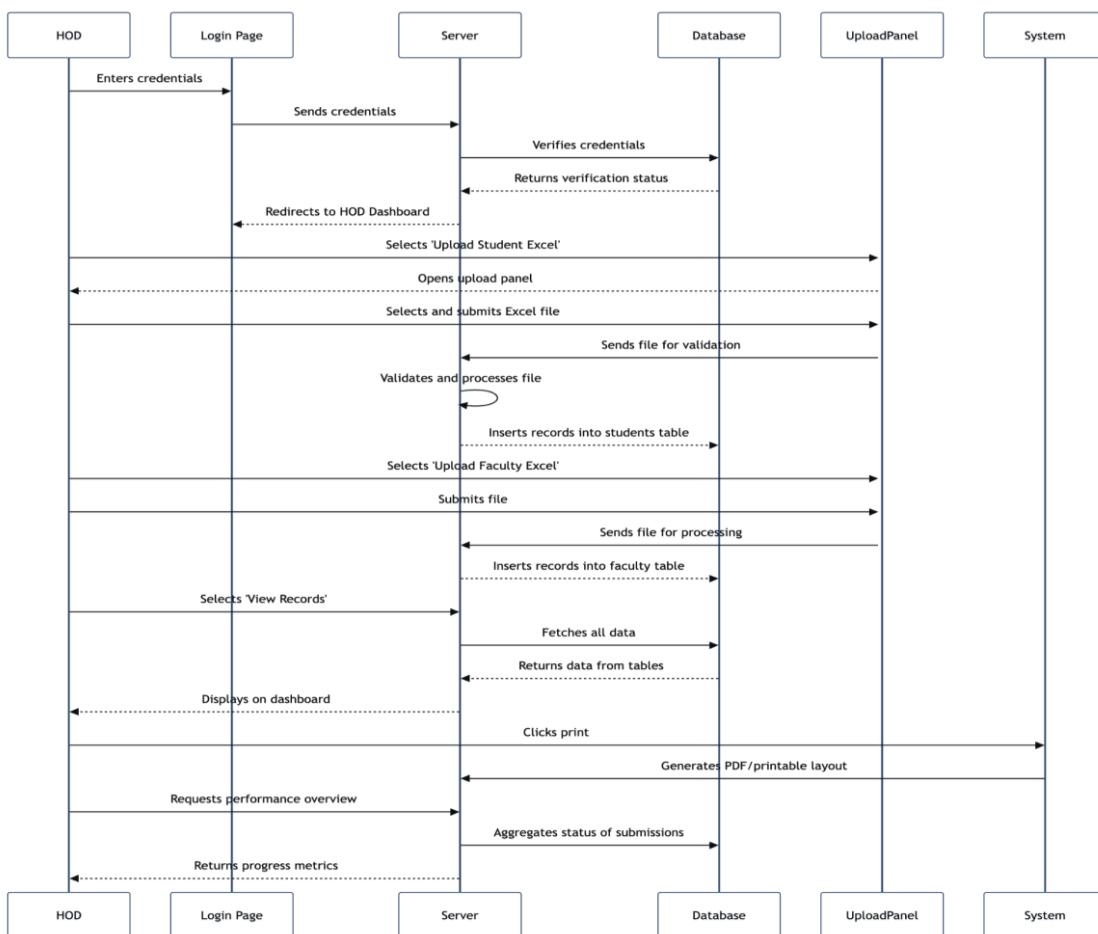


Fig 5.2.1 HOD Module Sequence Diagram

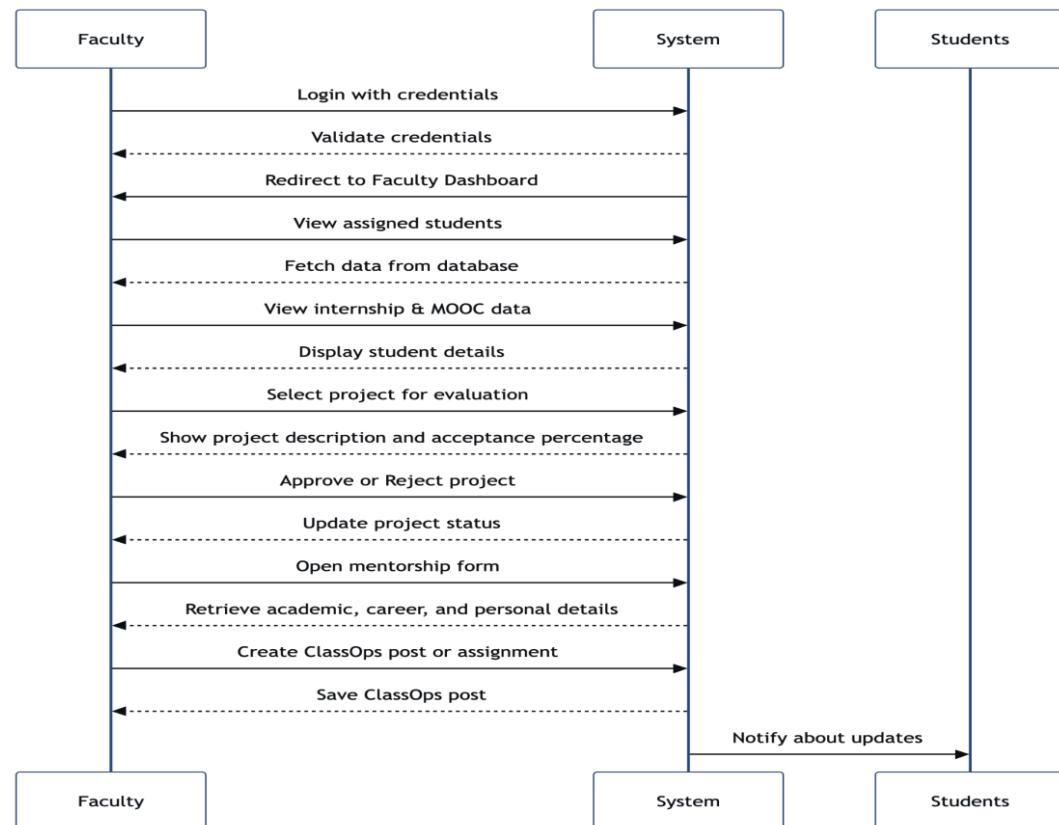


Fig 5.2.2 Faculty Module Sequence Diagram

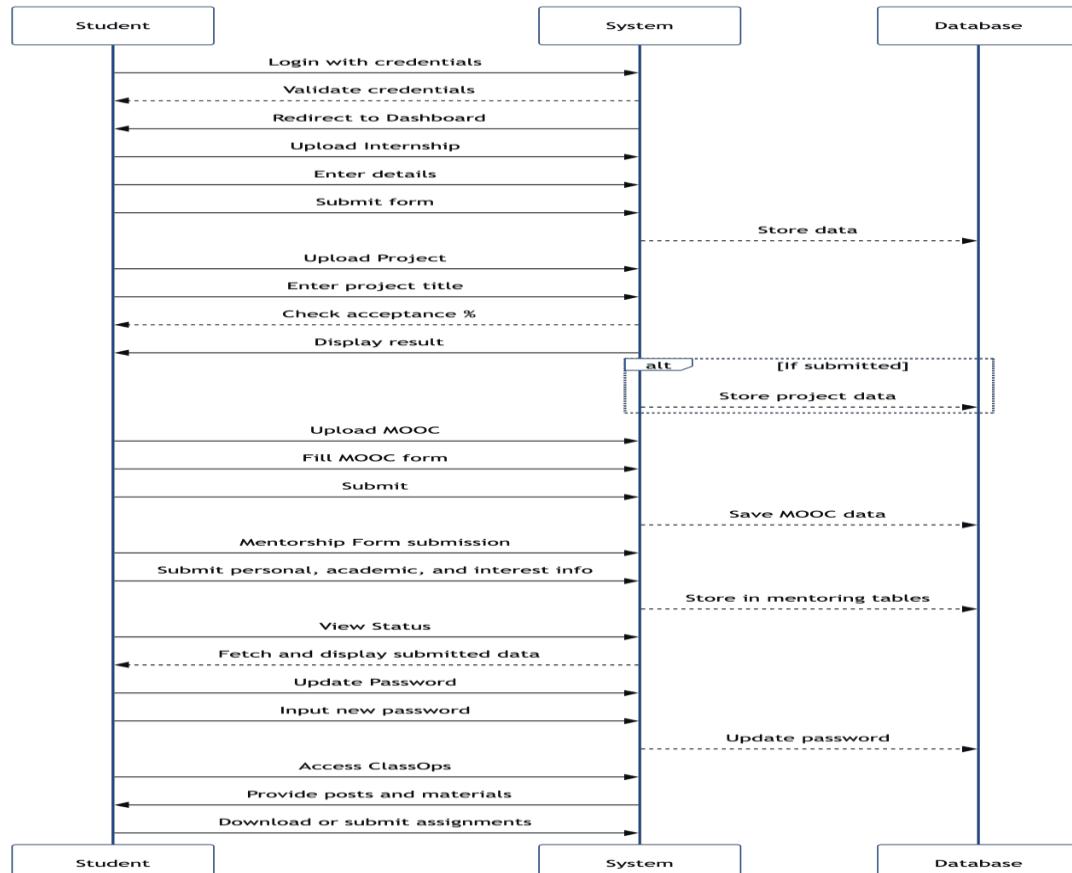


Fig 5.2.3 Student Module Sequence Diagram

The diagram showcases the order of student actions like logging in, submitting forms, and

participating in ClassOps. It reflects how the system processes each step in response to student inputs.

## 5.3 ACTIVITY DIAGRAM

The activity diagram illustrates the high-level dynamic behavior of the Learn Sync system, showcasing how users interact based on their roles—Student, Faculty, and HOD. It visualizes the control flow from login to respective dashboard functionalities and system operations.

### Description of Flow:

#### Common Entry Point:

- All users begin by logging into the system via a shared login interface.
- Based on the authenticated role, control is directed to the respective workflow: Student, Faculty, or HOD.

#### Student Workflow:

1. After login, students access their personalized dashboard.
2. They can upload:
  - Internship details
  - Project titles (with auto-evaluation for acceptance %)
  - MOOC certifications.
3. Students submit the Mentorship Form, providing academic, career, and personal details.
4. They can also:
  - Check the project acceptance percentage.
  - Access the ClassOps module for class updates and tasks.
  - View real-time project approval status from faculty.

#### Faculty Workflow:

1. Faculty logs in and accesses their dashboard to manage assigned students.
2. They can:
  - View internship, MOOC, and mentorship data of their students.
  - Evaluate projects and approve/reject proposals.
  - Post materials, announcements, or assignments via ClassOps.

#### HOD Workflow:

1. HODs upload bulk student and faculty data through Excel sheets.
2. They can view and print academic reports.
3. The HOD also monitors department-wide performance data and submissions across all modules.

This activity diagram represents a unified system with clearly defined, role-based paths

ensuring usability, control, and functionality. It effectively highlights how Learn Sync facilitates academic operations for every stakeholder.

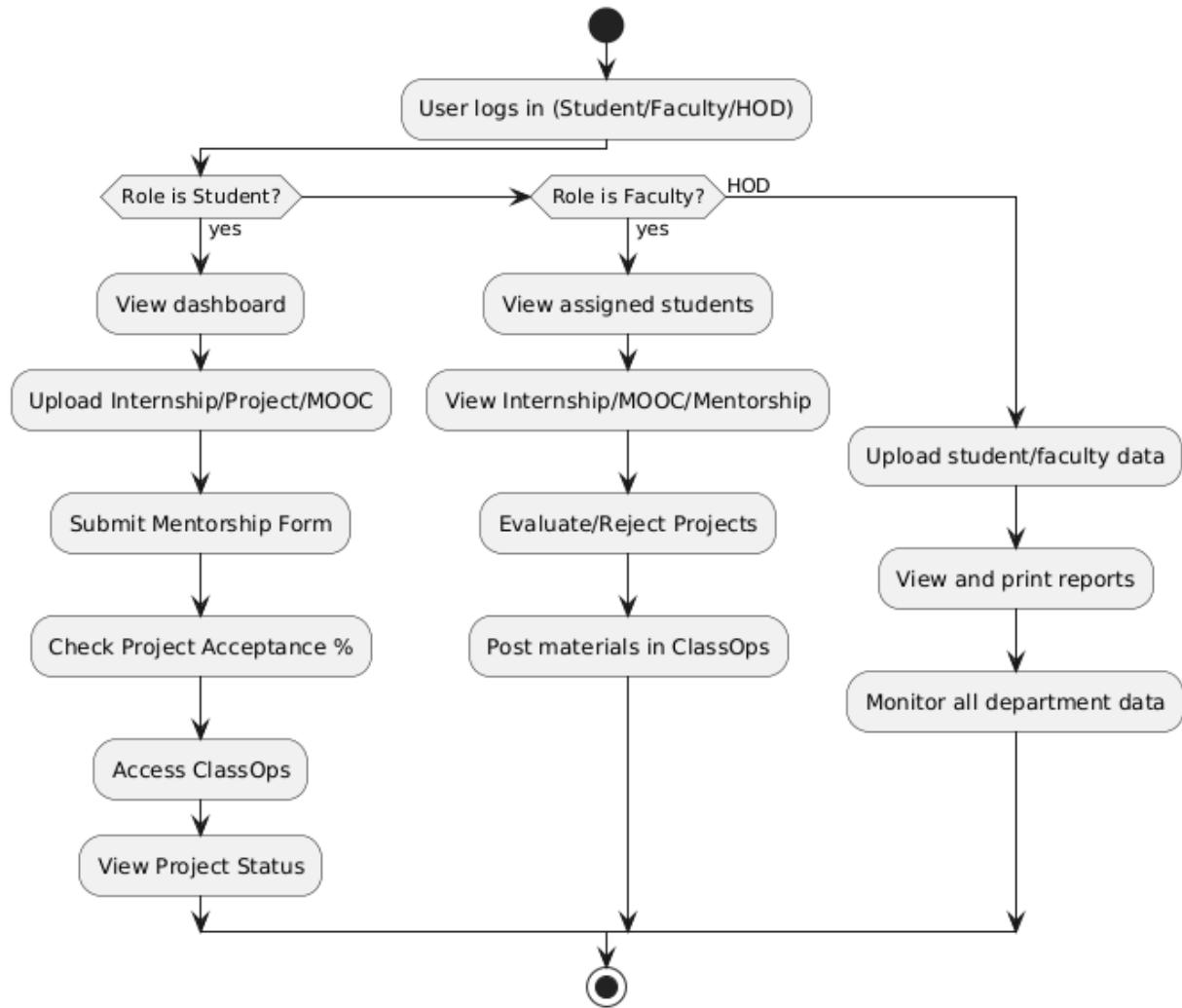


Fig 5.3.1 Activity Diagram

This activity diagram demonstrates the decision flow and role-based branching logic after login. It maps how students, faculty, and HODs proceed through different modules based on access rights.

## 5.4 DATABASE DESIGN

The Learn Sync system is backed by a structured and relational database designed to manage diverse academic modules while ensuring scalability, data integrity, and role-based access. The database uses MySQL and follows third normal form (3NF) to minimize redundancy and maintain consistency. Key objectives of the database design are:

- Support modular structure (Internship, Project, MOOC, Mentorship, ClassOps).
- Ensure referential integrity between tables through foreign key relationships.
- Allow secure storage of sensitive data (certificates, login credentials, evaluation results).

- Enable dynamic reporting and fast data retrieval.

## Core Tables and Their Functions:

### a) **students**

- Contains primary student details like USN, name, email, and login credentials.
- Acts as a reference for multiple modules.

### b) **faculty**

- Stores employee information for faculty members.
- Used in ClassOps and project evaluation.

### c) **internships**

- Holds data on student internships including role, location, dates, and certificate path.
- Linked to students.usn.

### d) **projects**

- Stores project titles, keywords, status, and rejection probability.
- Reviewed and updated by faculty.

### e) **mooc\_courses**

- Records details of online courses completed by students.
- Includes course name, provider, and certificate file path.

### f) **mentor\_form**

- Captures mentorship data including aspirations, email, phone, and submitted time.
- One-to-one mapping with students via usn.

### g) **student\_hobbies** and **student\_achievements**

- Support many-to-one relationships with mentor\_form for capturing multiple hobbies or achievements per student.

### h) **internal\_marks**

- Tracks semester-wise marks for Internal 1 and 2 per subject.
- Indexed by semester, internal number, and subject code.

### i) **classops\_posts**

- Enables faculty to post materials, announcements, and assignments.
- References faculty via employee\_id.

### j) **classops\_submissions**

- Stores student submissions (files) for ClassOps posts.
- Linked to both students.usn and classops\_posts.id

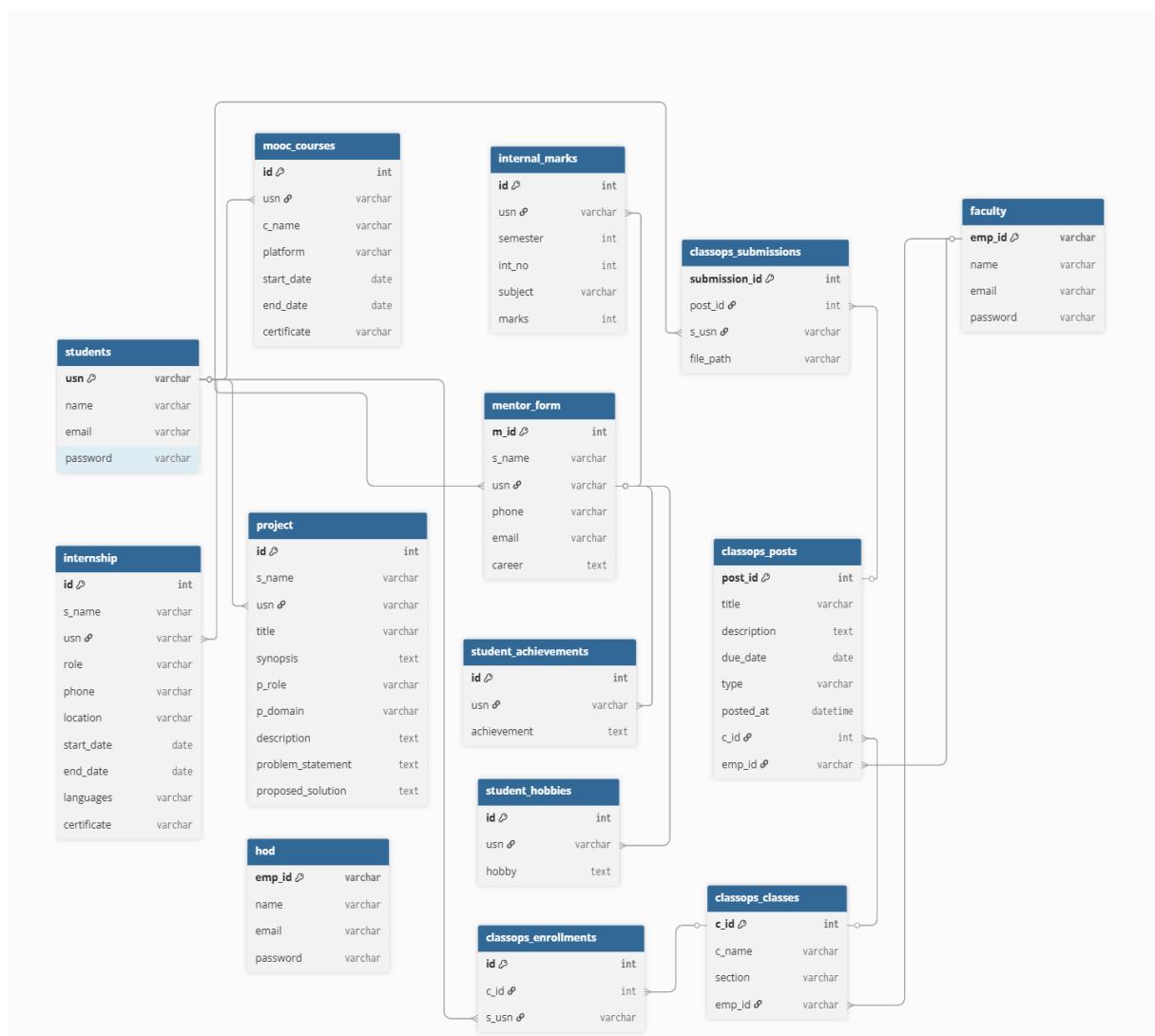


Fig 5.4.1 Database Schema Diagram

The database schema diagram outlines the table structures used in Learn Sync and how they relate to each other through foreign keys. It ensures organized data storage and enables efficient data retrieval.

#### 5.4.1 ENTITY-RELATIONSHIP (ER) DIAGRAM

The Entity-Relationship (ER) Diagram provides a high-level conceptual view of the Learn Sync database schema. It illustrates the data structure, interrelationships among key entities, and supports both developers and database administrators in understanding the logical organization and integration of academic modules.

##### Key Entities and Relationships:

**5.4.1.1 The students table** uses usn as its primary key and is linked to multiple other tables.

- Each student can submit multiple internships, so internships.usn references students.usn in a one-to-many relationship.
- A one-to-one relationship exists between students.usn and projects.usn, as each student can have only one active project.
- The mooc\_courses.usn also points to students.usn in a one-to-many relationship, since students can enroll in multiple MOOCs.
- The mentorship form (mentor\_form.usn) links directly to students.usn in a one-to-one mapping.
- Students can submit multiple responses to classroom activities, so classops\_submissions.student\_usn refers to students.usn.

#### **5.4.1.2 The faculty table** uses employee\_id as its primary key.

- Faculty members can create multiple ClassOps posts, establishing a one-to-many relationship where classops\_posts.faculty\_emp\_id references faculty.employee\_id.

#### **5.4.1.3 The internships table** contains details of student internships.

- The foreign key internships.usn ensures that every internship record belongs to a valid student in the students table.

#### **5.4.1.4 The projects table** stores project titles, descriptions, domains, and evaluation data.

- The column projects.usn references students.usn, ensuring that each project is associated with exactly one student.

#### **5.4.1.5 The mooc\_courses table** tracks completed MOOC certifications by students.

- The field mooc\_courses.usn connects each course record to a valid student in the students table.

#### **5.4.1.6 The mentor\_form table** collects mentorship inputs such as hobbies, aspirations, and internal marks.

- The primary key mentor\_form.usn also serves as a foreign key linking to students.usn, indicating that each student has one mentorship record.

#### **5.4.1.7 The student\_hobbies and student\_achievements tables** are connected to the mentor\_form table.

- Both use student\_hobbies.usn and student\_achievements.usn to reference mentor\_form.usn, establishing a many-to-one relationship.

#### **5.4.1.8 The internal\_marks table** holds semester-wise internal marks for each student.

- The column internal\_marks.usn links to mentor\_form.usn, showing that internal scores

are part of the overall mentorship record.

**5.4.1.9 The classops\_posts table** represents classroom posts and announcements made by faculty.

- The foreign key classops\_posts.faculty\_emp\_id ensures the post is assigned to the correct faculty member from the faculty table.

**5.4.1.10 The classops\_submissions table** records student responses to assignments and tasks posted in ClassOps.

- It contains two foreign keys:
  - classops\_submissions.post\_id → classops\_posts.id, linking submission to its respective post.
  - classops\_submissions.student\_usn → students.usn, ensuring submission is from a valid student.

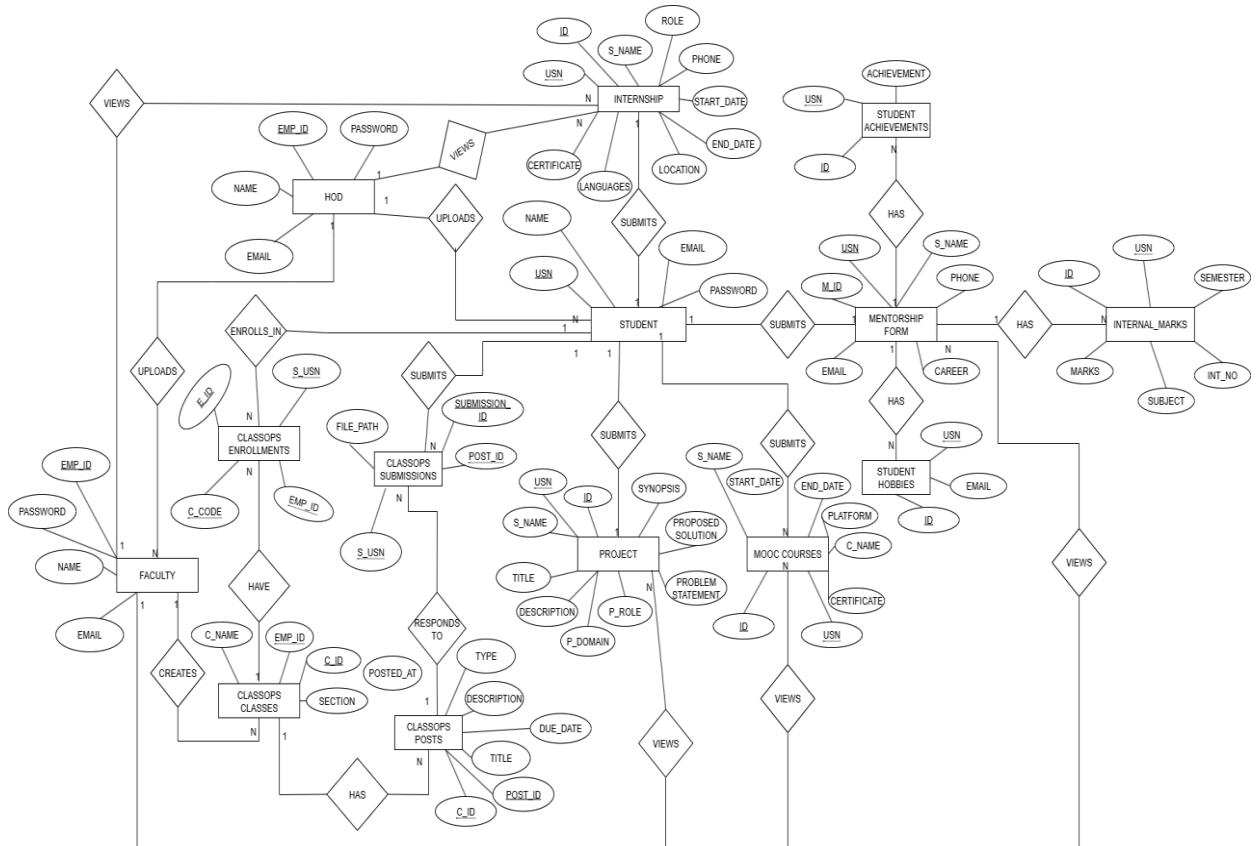


Fig 5.4.2 Entity Relationship Diagram

This ER diagram provides a conceptual model of how entities such as students, faculty, internships, projects, and mentorship data are interconnected. It helps understand the data relationships and enforces referential integrity.

# CHAPTER 6

## IMPLEMENTATION

### 6.1 INTRODUCTION

The implementation phase of *Learn Sync* focuses on converting the design and database architecture into a working, functional system using modern web technologies. It brings together user interface designs, backend logic, and database interactions to form a cohesive and responsive academic management platform. Implementation followed a modular approach - each module (Internship, Project, MOOC, Mentorship, and ClassOps) was developed and tested independently before integration to ensure system reliability and ease of debugging. Each user role (Student, Faculty, HOD) was provided with tailored functionalities through dashboards, ensuring secure access and task-specific operations. Server-side validation, database operations, and dynamic rendering were all handled using PHP and MySQL, while frontend interactivity was maintained using HTML, CSS, and JavaScript with Bootstrap and Tailwind CSS for responsive design.

### 6.2 PSEUDO CODES

Below are sample pseudocode snippets representing core logic flows of major modules in the system:

#### 1. Student Login Verification

Input: email, password

Process:

Fetch student data from database using the given email

If email exists:

    If password matches:

        Redirect to student dashboard

    Else:

        Show "Incorrect password" message

Else:

    Show "Email not found" message

## 2. Internship Submission

Input: internship form data, uploaded certificate

Process:

Validate form fields (start\_date, end\_date, location, etc.)

Check file format and size of certificate

Move certificate to server directory

Insert internship details and certificate path into 'internships' table

Output:

Show success message on completion

## 3. Project Title Evaluation

Input: student\_usn, project\_title

Process:

Extract keywords from title

Compare keywords with rejected\_projects table

Calculate rejection score and derive acceptance percentage

Display acceptance status

If title is submitted:

Insert record into 'projects' table with initial status as 'Pending'

## 4. Mentorship Form Submission

Input: student\_usn, career\_goal, hobbies, internal\_marks[]

Process:

Validate and sanitize inputs

Insert career details into 'mentor\_form' table

For each hobby:

Insert into 'student\_hobbies' table

For each subject's internal marks:

Insert into 'internal\_marks' table with usn and semester reference

## 5. Faculty Project Evaluation

Input: selected\_project\_id, decision (Approve/Reject)

Process:

Fetch project details from 'projects' table

- Faculty manually reviews title
- Update project status based on faculty decision
- Notify student of the updated status

## 6. ClassOps Post Creation (Faculty)

Input: post\_title, post\_description, task\_file

Process:

- Validate file format and content
- Insert post details into 'classops\_posts' table
- If file is uploaded:
  - Move file to storage and store path in database
  - Notify assigned students about new post

These pseudocodes reflect the logical design of backend operations and interactions among system components. All modules are designed to ensure data integrity, user-friendly navigation, and secure operations across roles.

## CHAPTER 7

# SOFTWARE TESTING

### 7.1 INTRODUCTION

Software Testing is a critical phase in the development lifecycle of Learn Sync, aimed at validating each module's correctness, completeness, performance, and reliability. The platform handles several academic operations such as internship tracking, project submission with predictive analysis, MOOC course entries, mentorship data, and classroom tasks - each needing rigorous verification to ensure seamless usability and functional accuracy.

Testing was conducted in stages, starting with unit-level validations and progressing toward integrated testing across modules. Both manual testing and logic-based automation checks were used to evaluate critical processes like form submissions, file uploads, access restrictions, and database integrity. The goal was to proactively detect errors, ensure data flow consistency, and deliver a stable academic management platform suitable for real-time institutional use.

### 7.2 TESTING OBJECTIVE

The primary objective of testing in Learn Sync was to:

- Verify the functionality and flow of all modules: Internship, Project, MOOC, Mentorship, and ClassOps.
- Validate that data entered by users (students, faculty, HOD) is stored and retrieved correctly.
- Ensure security measures such as role-based access are enforced properly.
- Confirm performance under normal academic workload.
- Identify and fix bugs, edge cases, and validation issues before deployment.

### 7.3 UNIT TESTING

Unit testing was performed module-by-module to ensure each functional unit of the system performs as expected. These isolated tests focus on:

- **Login and Role Validation:** Ensures correct redirection and error messages based on credentials.
- **Internship Module:** Tests data input, file upload handling, and database insertion.
- **Project Module:** Validates title analysis logic and prediction percentage accuracy.

- **MOOC Module:** Checks MOOC details submission and certificate viewing.
- **Mentorship Module:** Tests multi-entry fields like hobbies, internal marks, and data grouping per semester.
- **ClassOps Module:** Verifies faculty posts, file attachments, and student submission flows.
- **Search and Filter Features:** Ensures data is dynamically filtered by USN or name in real time.
- **Data Uploads:** Validates Excel upload functionalities and database integration by HOD.

Each unit was tested with valid, invalid, and boundary inputs to ensure the robustness of validation logic, form behaviour, and database operations.

### 7.3.1 UNIT TESTING TEST CASES

**Table 7.3.1.1: Testing for Login & Authentication Module**

Test Case ID	Description	Input Data	Expected Output	Status
L001	Login with valid credentials	Email: <a href="mailto:student@sjec.ac.in">student@sjec.ac.in</a> Password: Student@123	Redirect to student dashboard	Pass
L002	Login with incorrect password	Valid email, wrong password	Show "Incorrect password"	Pass
L003	Login with non-existing email	Email not in Database	Show "Email not found"	Pass
L004	Password change	New password meets validation	Password updated and encrypted	Pass
L005	Access control based on role	Faculty logs in	Redirected to faculty dashboard	Pass
L006	Login with non SJEC Email	Email: faculty@example.com	Show "Enter valid email address"	Pass

This table validates the login functionality across different roles (Student, Faculty) by testing credential handling, role-based redirection, and password change. It ensures only authorized users with valid SJEC emails can access the system securely.

**Table 7.3.1.2: Testing For Internship Module**

Test Case ID	Description	Input Data	Expected Output	Status
I001	Submit internship form	Valid form with all fields	Record saved in database	Pass
I002	Upload valid certificate	PDF/JPG file	File saved and path stored	Pass
I003	Upload unsupported file format	.exe file	Show "Invalid file format"	Pass
I004	Submit with empty required fields	Missing phone number	Form not submitted, alert shown	Pass
I005	View internship details	Click view	Display of internship with certificate link	Pass
I006	Upload file > 200KB	Choose File	Show "File size should be < 200KB"	Pass

This table tests the functionality of the internship form, including field validations, file format restrictions, and certificate upload limits. It ensures proper data storage and user feedback for successful or invalid submissions.

**Table 7.3.1.3: Testing for Project Module**

Test Case ID	Description	Input Data	Expected Output	Status
P001	Submit project title	Title: "AI Resume Evaluator"	Acceptance % shown, data saved	Pass
P002	Title with common rejected terms	"E-commerce Website"	Lower acceptance % displayed	Pass
P003	Upload details multiple times	Click "Enter Details"	Show "details already submitted"	Pass

P004	Empty title field	—	Show "Field required" message	Pass
------	-------------------	---	-------------------------------	------

This table verifies the project title evaluation system by testing title submission, rejection logic based on keywords, and duplicate entry handling. It ensures students receive immediate feedback and prevents resubmission of the same data.

**Table 7.3.1.4: Testing For MOOC Course Module**

Test Case ID	Description	Input Data	Expected Output	Status
M001	Submit MOOC details	Platform: Coursera + Certificate	Entry stored in database	Pass
M002	Submit without certificate	Missing file	Show "Certificate required"	Pass
M003	Upload multiple times	Multiple certificates	Store in Database & Display all the under same USN	Pass
M004	Filter MOOC by name	"Ajay"	Matching records displayed	Pass

This table evaluates the MOOC module's ability to handle submissions, validate file uploads, and support multiple entries per student. It also tests the search functionality to retrieve specific records by student name.

**Table 7.3.1.5: Testing For Mentorship Module**

Test Case ID	Description	Input Data	Expected Output	Status
MT001	Submit mentorship form	Name, USN, Email, etc.	Data saved into mentor_form	Pass
MT002	Add hobbies	2 hobbies entered	Saved in student_hobbies	Pass
MT003	Add internal marks for 3 subjects	Sem 1, Internal 1	Records saved in internal_marks	Pass
MT004	Update existing mentorship record	Changed aspirations only	Only modified data updated	Pass

This table checks the mentorship form's ability to handle student data, including hobbies and internal marks. It also validates partial updates, ensuring only modified fields are updated without affecting existing records.

**Table 7.3.1.6: Testing For ClassOps Module**

Test Case ID	Description	Input Data	Expected Output	Status
C001	Faculty posts announcement	Valid title, description	Stored in classops_posts	Pass
C002	Student views class post	Logged-in student	Post visible in student dashboard	Pass
C003	Assignment file submission	Valid PDF uploaded	Stored in classops_submissions	Pass
C004	View uploaded assignment	Faculty clicks on submission	File preview or download prompt	Pass
C005	Post filtering by title	Search: "Unit 1 Notes"	Matching post displayed	Pass

This table validates the ClassOps module's core features including faculty post creation, student views, assignment submissions, and post search functionality. It ensures smooth interaction between faculty and students within the digital classroom environment.

**Table 7.3.1.7: Testing For Bulk Upload (HOD Module)**

Test Case ID	Description	Input Data	Expected Output	Status
H001	Upload student Excel file	.xlsx File with 120 students	Records inserted to student table	Pass
H002	Upload file with duplicates	Existing USNs in Excel	Duplicates skipped and shown as alert	Pass
H003	Upload malformed Excel	Corrupted or incorrect columns	Show upload error message	Pass
H004	View uploaded data	Go to student view page	Newly uploaded records visible	Pass

This table tests the bulk upload functionality, ensuring accurate import of student records from Excel files. It handles duplicates, malformed data, and confirms that uploaded records are correctly reflected in the system.

## 7.4 INTEGRATION TESTING

Integration testing is a crucial phase in the validation process of the Learn Sync system. After individual modules were verified through unit testing, integration testing was conducted to ensure that the combined modules worked together seamlessly as a complete system. This step helped detect issues related to data flow, communication between components, user role transitions, and multi-module interactions.

The primary goal was to test the functional correctness and consistency of workflows involving multiple components—such as student form submissions, faculty reviews, dashboard updates, and certificate uploads. It ensured that form inputs, file handling, backend logic, and database operations were correctly connected and executed as intended.

Particular emphasis was placed on modules such as Internship, Project Evaluation, MOOC, Mentorship, ClassOps, and Bulk Upload, all of which required smooth communication between the frontend, backend, and database layers. The tests also verified session management, data visibility across roles (Student, Faculty, HOD), and real-time updates without any data loss or misrouting.

By executing a series of real-time academic scenarios, integration testing confirmed that Learn Sync functions reliably in a realistic multi-user environment, ensuring end-to-end academic process automation with minimal manual intervention.

### 7.4.1 INTEGRATION TESTING TEST CASES

**Table 7.4.1.1: Integration Testing Test Cases**

Test Case ID	Module(s) Involved	Test Scenario	Test Data	Expected Result	Actual Result	Status
INT01	Login + Role Routing	Verify that user is redirected to the correct dashboard based on their role	Student login, Faculty login, HOD login	Role-based redirection works appropriately	As Expected	Pass
INT02	Internship + File Upload	Student uploads internship details and a certificate	Internship data + PDF file	Data saved in database, certificate viewable for faculty & HOD	As Expected	Pass
INT03	Internship + View	Faculty views student internship	Faculty access to student record	Accurate details displayed with certificate in	As Expected	Pass

		details		modal		
INT04	Project + Prediction	Student submits project title and system returns prediction percentage	Title: "E-Commerce Website"	Prediction percentage shown based on keyword analysis	As Expected	Pass
INT05	Classops assignment submission and grading	Student logs in and submits assignment and faculty grades the submissions	Project entry exists in DB	Title and prediction result displayed in student dashboard	As Expected	Pass
INT06	MOOC Certificate Upload + Display	Student uploads MOOC details with certificate for 2 courses	Course: NPTEL, Coursera	Mooc Details with Certificates of both courses	Data and certificate stored; accessible for faculty	As Expected
INT07	MOOC + Faculty View	Faculty views submitted MOOC course of assigned students	Faculty logs in	MOOC entries visible under assigned student profiles	As Expected	Pass
INT08	Mentorship + Form Submission	Student fills mentorship form including hobbies and aspirations	Form data, hobbies: [Coding, Writing]	Records saved in mentor_form and student_hobbies tables	As Expected	Pass
INT09	Mentorship + Internal Marks	Student adds internal marks, faculty views complete mentorship data	Sem 2, Internal 1, Subject: ADA, Marks: 21	Internal marks saved and visible to faculty in mentorship view	As Expected	Pass
INT10	ClassOps + Post + Submission	Faculty posts assignment; student submits assignment	Post: "Assignment 1"; Submission: PDF file	Post visible to students; submission stored in classops_submissions table	As Expected	Pass
INT11	ClassOps + Student Dashboard	Student checks ClassOps dashboard and downloads class materials	Student logs in	All faculty posts for enrolled classrooms visible	As Expected	Pass
INT12	HOD + Bulk Student Upload	HOD uploads Excel file with student data	Valid .xlsx file	Data inserted to students table with duplicate entries skipped	As Expected	Pass
INT13	HOD +	HOD views	Dept.: MCA	Data filtered and	As	Pass

	Data View & Print	uploaded records and prints filtered reports of their department only	Use filters: “Internship”	printable format generated	Expected	

This table verifies that integrated components of Learn Sync function cohesively when interacting across modules. It checks cross-module scenarios such as login + dashboard routing, internship uploads + faculty views, and mentorship form + internal marks, ensuring reliable data flow and end-to-end functionality.

## CHAPTER 8

### CONCLUSION

The Learn Sync project has effectively addressed the need for a centralized and efficient academic management system designed for higher education institutions. It combines various academic workflows, such as internship tracking, project evaluation, MOOC certificate management, mentorship records, and digital classroom operations, into a single platform. This integration simplifies data submission and management. It also removes the inefficiencies and inconsistencies tied to traditional manual systems like paper forms and spreadsheets.

By adding smart automation features, such as AI-based project title evaluation and bulk uploads from Excel, Learn Sync improves accuracy and reduces the administrative workload. Each user role, like students, faculty, and HOD, has a specially designed interface that provides required access, functionality, and ease of use. Faculty can access mentorship records, monitor student engagement in MOOCs and internships, and manage classroom content easily. At the same time, students can get a straightforward submission process and tailored academic tracking, while HODs can oversee their departments and create printable reports for academic audit and compliance.

From a technical perspective, the system features a modular, role-based structure that supports maintenance and expansion. Its secure design protects data integrity and access control, while the layered approach allows for future additions or customizations by the departments.

In summary, Learn Sync meets the current academic management needs and sets the stage for a smart, digital-first academic environment. Its implementation provides a model for other institutions looking to digitize and improve student engagement, faculty collaboration, and administrative oversight through a single platform.

## CHAPTER 9

### FUTURE ENHANCEMENTS

While Learn Sync already delivers comprehensive academic process management across departments, additional improvements can elevate its usability, scalability, and automation. Here are the potential enhancements planned for future versions of the system:

- 1. Student Performance Analytics Dashboard:** Integrate data visualization tools to show academic trends, semester-wise performance, and activity heatmaps. This dashboard will help students and mentors make informed decisions. Insights from mentorship and course participation will also be visualized.
- 2. Faculty Feedback and Evaluation System:** Add a proper feedback mechanism where students can evaluate their mentorship experience and faculty support. This system can assist HODs in reviewing teaching qualities and identifying areas for faculty development.
- 3. Mobile Application Support:** Design a lightweight Android and iOS application that mirrors dashboard functionalities. It will allow users to view records, upload files, receive notifications, and participate in classroom tasks seamlessly on the go.
- 4. Digital Signature and Approval Workflow:** Introduce secure digital signature capabilities for validating certificates, reports, and mentorship records. This will help eliminate the dependency on physical documentation, especially during audits or remote operations.
- 5. Calendar and Event Integration:** Incorporate a shared academic calendar with color-coded entries for internship deadlines, assignment submissions, mentorship meets, and exams. Users will get visual cues and reminders for better time management.
- 6. Offline Data Entry Mode:** Enable users to fill out forms and upload documents without needing an active internet connection. The system will store data locally and sync it to the server once the device is back online, ensuring accessibility in low-connectivity areas.
- 7. AI-Powered Mentorship Insights:** Use AI and natural language processing to analyse mentorship data like hobbies, aspirations, and academic performance. The system can then recommend personalized learning paths, certifications, or career suggestions.

## APPENDIX A

### REFERENCES

- [1] **St. Joseph Engineering College (SJEC) Website:** <https://www.sjec.ac.in> – Source for institutional profile, department structure, and accreditation details.
- [2] **MySQL Official Documentation:** <https://dev.mysql.com/doc/> – Reference for writing optimized queries, schema creation, and relational design.
- [3] **W3Schools & MDN Web Docs:** <https://www.w3schools.com> and <https://developer.mozilla.org> – Frontend development references for HTML, CSS, JavaScript, and responsive UI design.
- [4] **PHP Manual – Official Site:** <https://www.php.net/manual/en/> – Used for PHP integration with MySQL and handling form validations, sessions, and file uploads.
- [5] **Draw.io & Lucidchart:** Used for creating architectural diagrams, sequence diagrams, DFDs, and ER diagrams in the system design and detailed design phases  
<https://www.lucidchart.com> <https://www.diagrams.net/>  
<https://dev.mysql.com/downloads/workbench/>.
- [6] **IEEE Standards for Software Documentation:** Followed as the base format for preparing structured SRS, system design, and testing documentation.
- [7] **YouTube & Online Forums (Stack Overflow, GitHub Discussions):** Helpful in resolving issues related to backend debugging, project evaluation logic, and ClassOps module implementation.

## APPENDIX B

### USER MANUAL

#### LOGIN PAGE

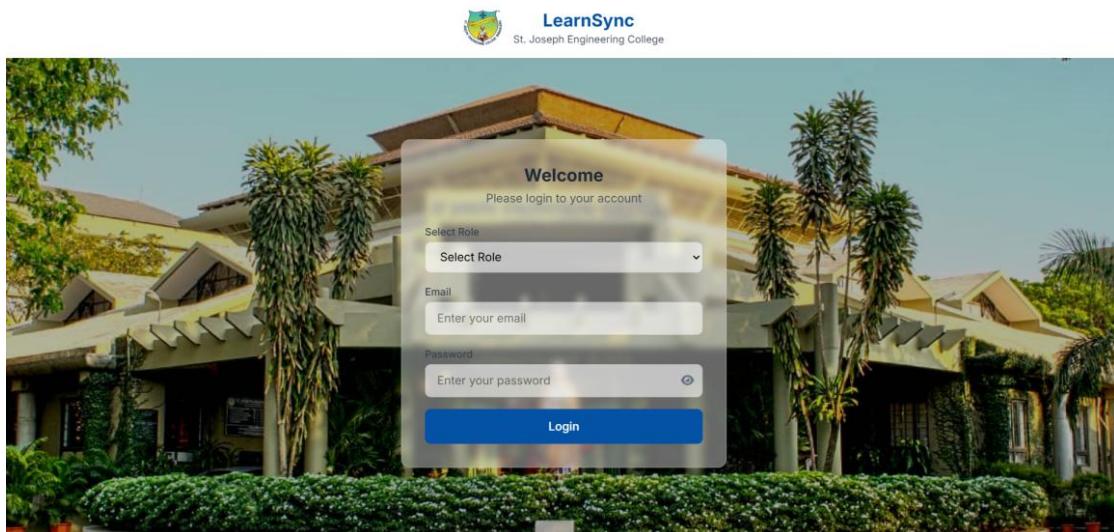


Fig. 1. Login Page

The login page serves as the entry point, allowing users to authenticate based on their role (Student, Faculty, or HOD). Secure input validation ensures only registered users gain access to their respective dashboards.

#### STUDENT DASHBOARD

Fig. 2. Student Dashboard

The student dashboard provides navigation to internship, project, MOOC, mentorship, and classroom modules. It displays a personalized interface with access to forms and uploaded data.

## INTERNSHIP DETAILS PAGE

Fig. 3. Internship Details Page

Students can submit internship details, upload certificates, and view their submissions. Validation ensures only proper formats are accepted, and submitted data is stored securely.

## PROJECT DETAILS PAGE

Fig. 4. Project Details Page

This page allows students to enter project titles for backend analysis. The system returns a project acceptance percentage, helping students assess feasibility before finalization.

## MOOC DETAILS PAGE

The screenshot shows the 'MOOC Course Details' page. At the top, there's a header with the LearnSync logo and navigation links for Home, Internship, Projects, Courses, Mentorship, Class Ops, and a user profile for Sushmitha. The main section has a blue header 'MOOC Course Details' with the sub-instruction 'Share your online learning experience with us'. Below this, a note says 'Fields marked with \* are required'. The 'Personal Information' section contains fields for Name\* (Sushmitha Monthero) and USN\* (4SO23MC107). The 'Course Details' section includes fields for Course Name\* (Name of the MOOC course), Platform\* (e.g., Coursera, edX, Udemy), Start Date\*, End Date\*, Certificate Link\*, and a file upload field for Certificate File\* (Accepted formats: PDF, DOC, DOCX, JPG, PNG (Max 200KB)). A large blue 'Submit Course Details' button is at the bottom.

Fig. 5. MOOC Details Page

Students enter MOOC course details and upload corresponding certificates. All submissions are tracked and stored, enabling faculty and HODs to verify course completions.

## MENTORSHIP DETAILS PAGE

The screenshot shows the 'Mentorship Details' page. The header is identical to the MOOC page. The main section has a blue header 'Mentorship Details' with the sub-instruction 'Share your academic and personal details with your mentor'. It is divided into several sections: 'Student Information' (Full Name\* (Sushmitha Monthero), USN\* (4SO23MC107), Email\* (23ca107.sushmitha@sjec.ac.in), Phone Number), 'Internal Marks' (Select Semester\*), 'Hobbies' (Enter hobby and Add Another Hobby), 'Achievements' (Enter achievement and Add Another Achievement), and 'Career Aspirations' (Describe career goals and aspirations...).

Fig. 6. Mentorship Details Page

This section captures students' personal and academic information including hobbies, aspirations, and internal marks. Submitted data helps faculty guide students more effectively.

## CLASSOPS DASHBOARD

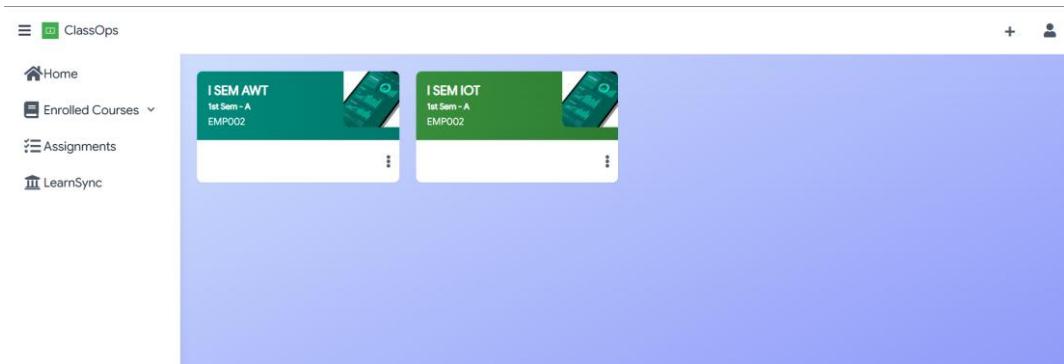


Fig. 7 Classops Dashboard

The ClassOps dashboard lists all enrolled classes and posts created by faculty. Students can view announcements, materials, and assignments relevant to their classrooms.

## ENROLLED CLASS PAGE

This screenshot shows the 'Enrolled Class Page' for the 'I SEM AWT' class. The page has a sidebar with the same navigation options as the dashboard. The main content area shows a post from 'Dr. Alice Taylor' with the title 'write answers to all the questions'. It includes an attachment 'Question Bank 3 &.pdf'. Below the post, under 'Your Work', is a submission entry for a student. The submission was submitted on 'Jun 27, 2025 7:37 PM' with the file 'Program 3.pdf' and a grade of '70'. The feedback states 'all the questions are not answered'. There is also an 'Edit submission' link. At the bottom, there is a 'Comments' section with a placeholder 'Add a comment...'.

Fig. 8. Enrolled Class Page

Displays all content shared within a specific classroom including faculty posts and uploaded resources. Students can interact with each post as per the task requirement.

## JOIN CLASS MODAL

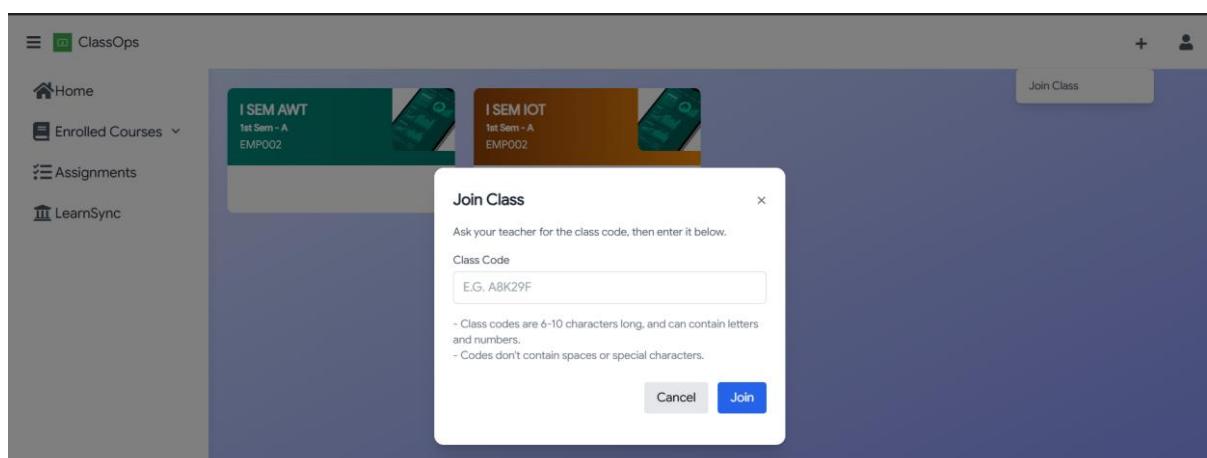


Fig. 9. Join Class Modal

Students can join a class using a class code provided by faculty. Validation checks ensure the class code is active and valid before adding the student to the class.

## CREATE CLASS MODAL

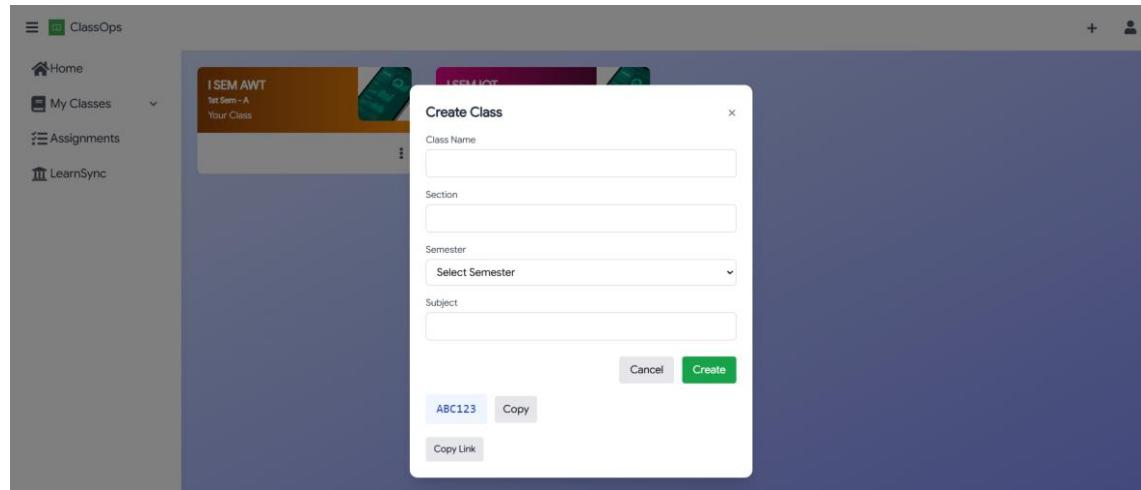


Fig. 10. Create Class Modal

Faculty can create a new classroom by specifying title, section, and subject. Upon creation, the class appears on the dashboard and is ready for post creation and student joining.

## CREATE CLASS POST PAGE

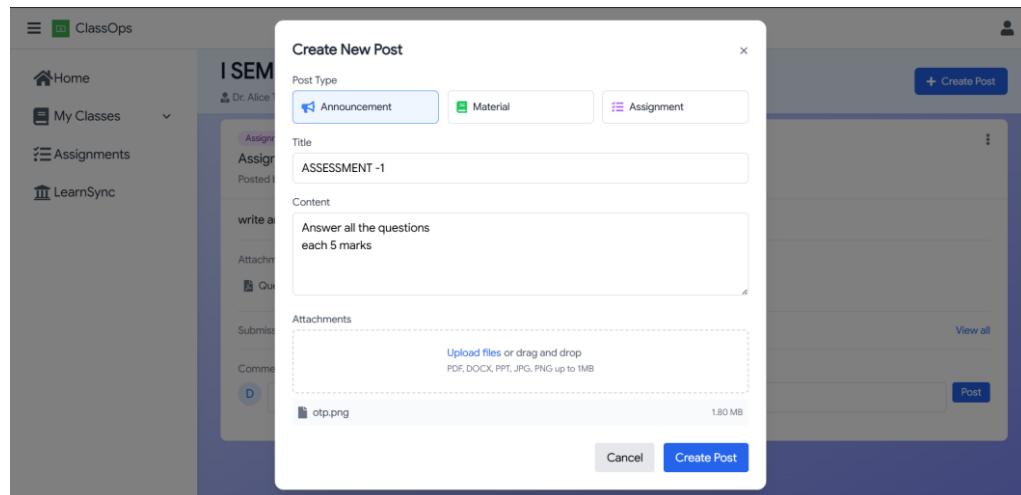


Fig. 11. Create Class Post Modal

Faculty use this page to post materials, announcements, or assignments. Attachments can be included, and each post becomes visible to all enrolled students.

## STUDENT ASSIGNMENT SUBMISSION PAGE

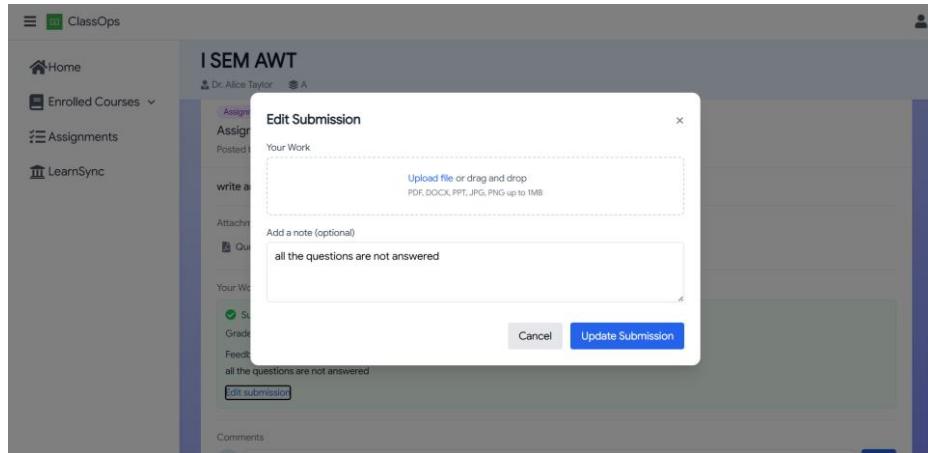


Fig. 12. Submit Assignment Page

Students submit assignment files in response to faculty posts. Submissions are saved in the system for later review and evaluation by the respective faculty member.

## UPDATE PASSWORD PAGE

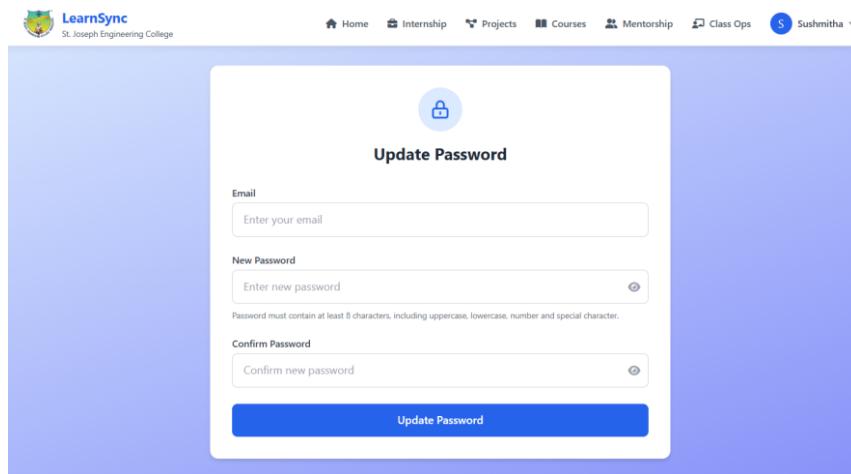


Fig. 13. Update Password Page

All users can update their login password through this page. Passwords are securely encrypted and validated before being updated in the database.

## FACULTY DASHBOARD

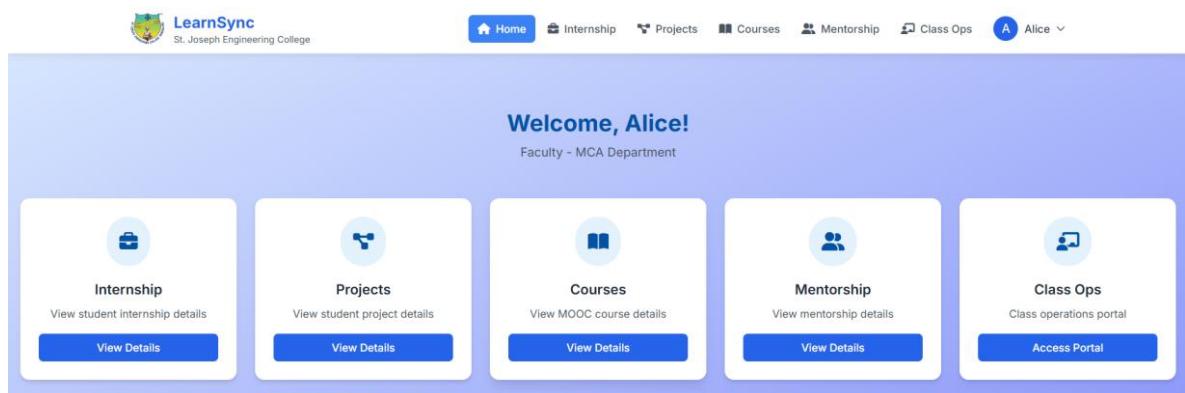


Fig. 14. Faculty Dashboard Page

Faculty access student records, view submitted project/MOOC/internship/mentorship details, and manage ClassOps. The dashboard is personalized for each faculty member.

## VIEW INTERNSHIP DETAILS PAGE

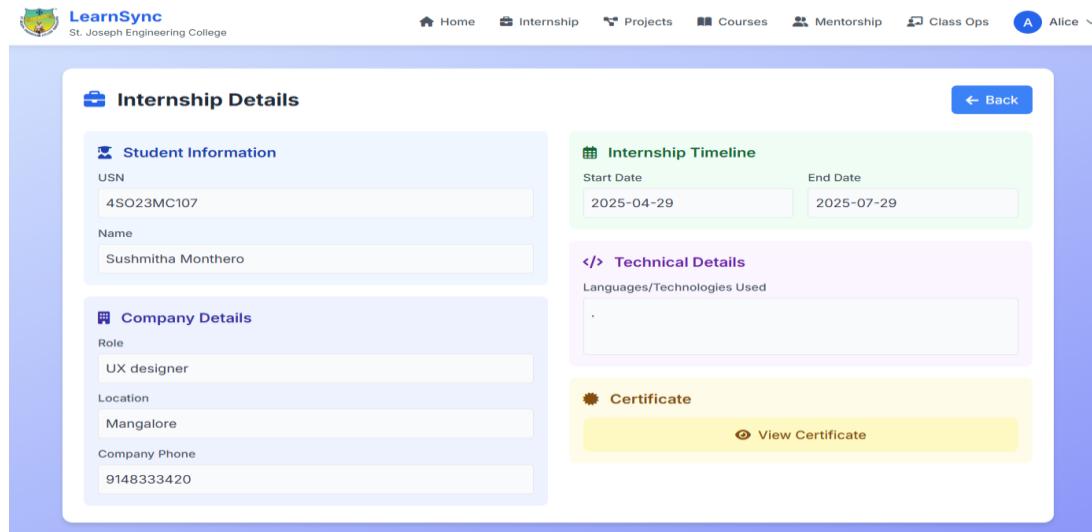


Fig. 15. View Internship Details Page

Faculty can view detailed internship submissions from students. A modal-based interface allows quick access to uploaded certificates without page reloads.

## VIEW CERTIFICATE DETAILS

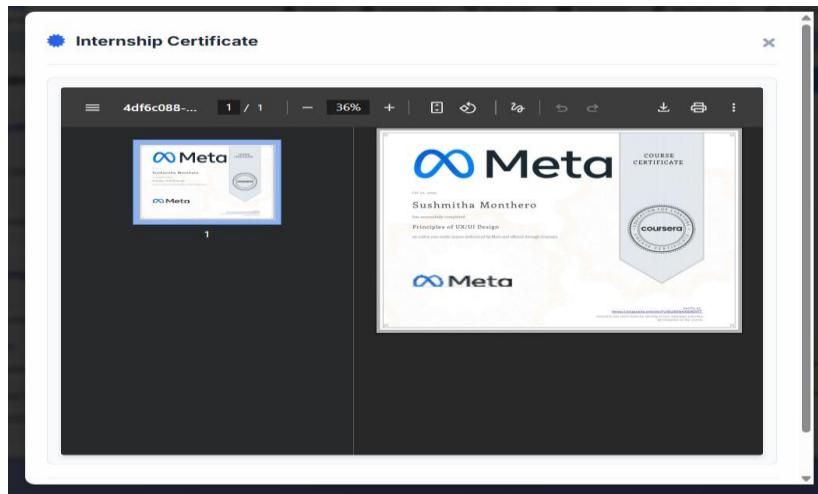


Fig. 15. View Certificate Modal

This modal displays internship or MOOC certificates uploaded by students. Faculty and HODs can verify authenticity directly from the dashboard interface.

## HOD DASHBOARD PAGE

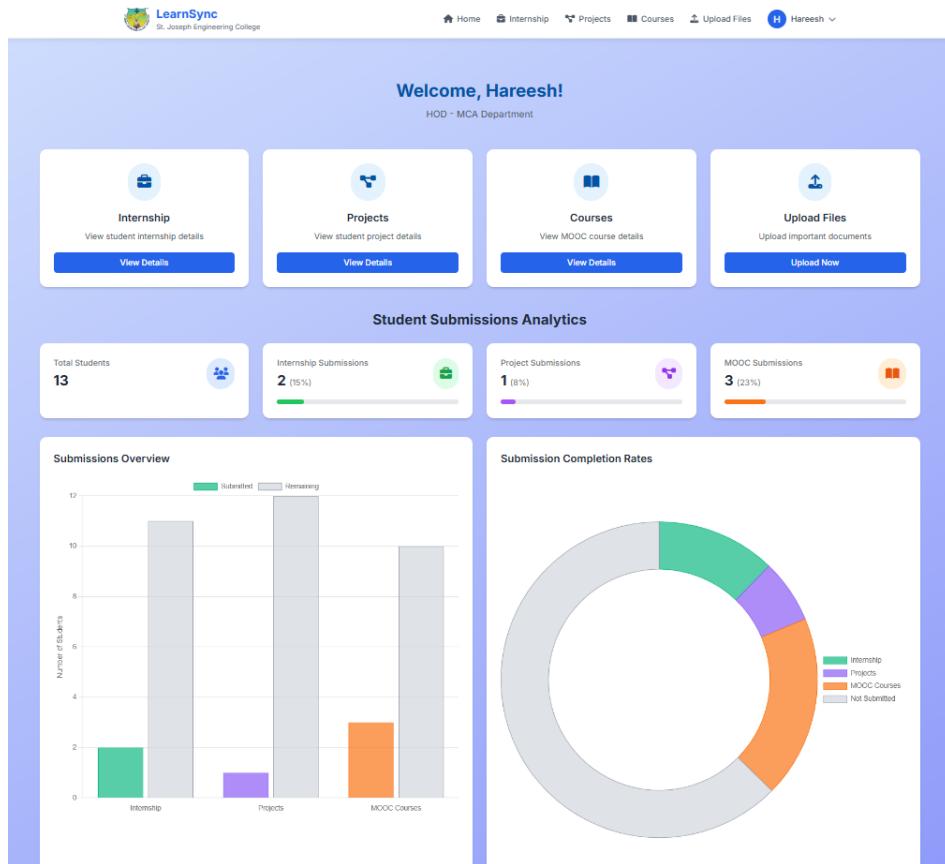


Fig. 16. HOD Dashboard

The HOD dashboard offers full administrative access across all modules. It allows HODs to view student data, upload files, and generate printable reports.

## UPLOAD FILES PAGE

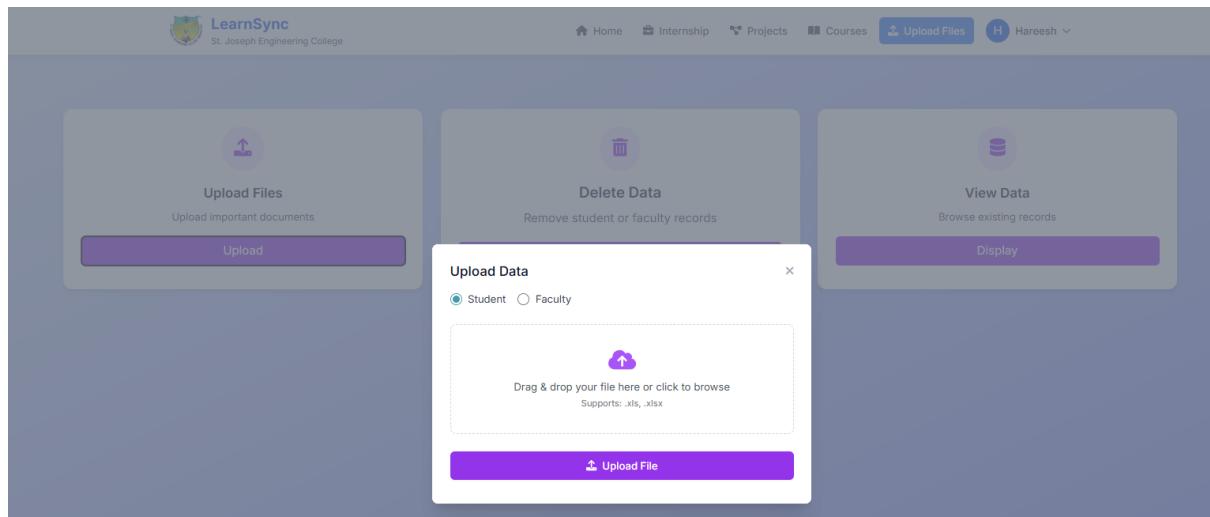


Fig. 17. HOD Upload Files Page

This page allows the HOD to upload Excel files containing student or faculty records. Uploaded data is validated and inserted into the database.

## VIEW USER DETAILS PAGE

Name	USN	Email	Department	Password
Preetham	4SO23MC001	preetham@sjec.ac.in	MCA	Mangalore@123
Dilan	4SO23MC002	dilan@sjec.ac.in	MCA	\$2y\$10\$80JNLq9CPKISXP3vs vQQuexVioAEoopeeCzR2vjVeL8mOu92Ny6Bq
Isthiyaq	4SO23MC003	preetham@sjec.ac.in	MCA	\$2y\$10\$!5BjqRGxSXU4KZdNi.m3eHYOCuooclHEzlyc2.bwaVd847F2rYBK
Anish	4SO23MC004	anish@sjec.ac.in	MCA	\$2y\$10\$MSRCBBedfIPxPuYzr77hdOyCXRWsgdAmeIYea/QD3Et1IJFp1iBY2
Gaurav	4SO23MC005	gaurav@sjec.ac.in	MCA	\$2y\$10\$42fr6NwmsPLm4.MZOPU.U/.pstJpCQk567uOA71tdR.T.5jmmZBC
Sherwin	4SO23MC006	sherwin@sjec.ac.in	MCA	\$2y\$10\$.azAg/qXQ7wlK2exKB0vNuWCd0nqCDijvk7q/z2ByFGJzwzldgcufi
Royston	4SO23MC007	royston@sjec.ac.in	MCA	\$2y\$10\$j7F39aS.lo61k/nBsBMHPOBe3WXr3J6r02EZEEnPvPm8dr3giMT8Lq
Francis	4SO23MC008	francis@sjec.ac.in	MCA	\$2y\$10\$JrvWKIhee4YeHy0m2vuhpuQThoXaiziVOKmVwNaY.4SX4.9kuoSa
Joyal	4SO23MC009	joyal@sjec.ac.in	MCA	\$2y\$10\$.hLp6dg7RK/JbFKSpBmWFu4/TEQLBlaA6jjQy7aEqkRaBs09HzpRW
Caysus	4SO23MC010	caysus@sjec.ac.in	MCA	\$2y\$10\$A3YdiuIEl0cOUNI4X6fuLhJQIO.uDuR6aOeL7rh6aw9PCdMBB4y
Canvil	4SO23MC016	canvil@sjec.ac.in	MCA	Mangalore@123
Jeffson Dsilva	4SO23MC036	jeffson@sjec.ac.in	MCA	Mangalore@123
Sushmitha Monthero	4SO23MC107	23ca107.sushmitha@sjec.ac.in	MCA	Mangalore@123

Fig. 18. HOD View Student Records Page

HOD can view detailed records of all students across departments. Filtering and search options enable quick access to specific entries.

## PROJECT PREDICTION MODEL

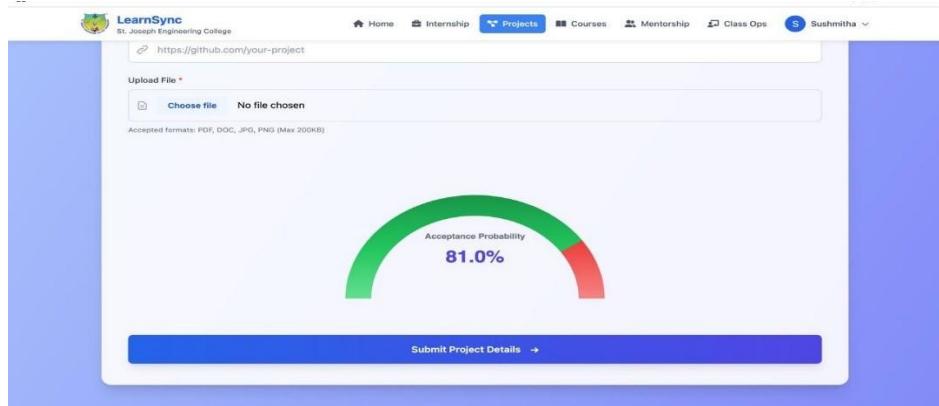


Fig. 19. Project Prediction Model

This component evaluates submitted project titles using keyword-based logic and historical rejection data. It provides students with an acceptance percentage to guide title selection.



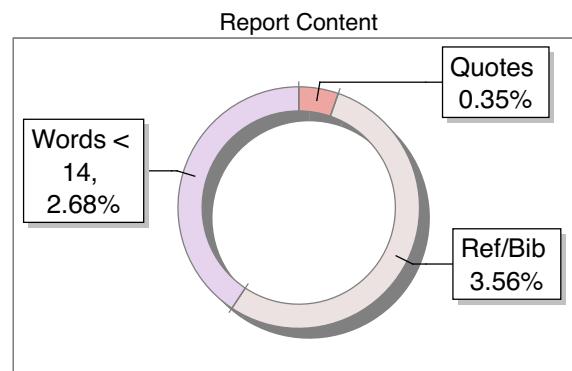
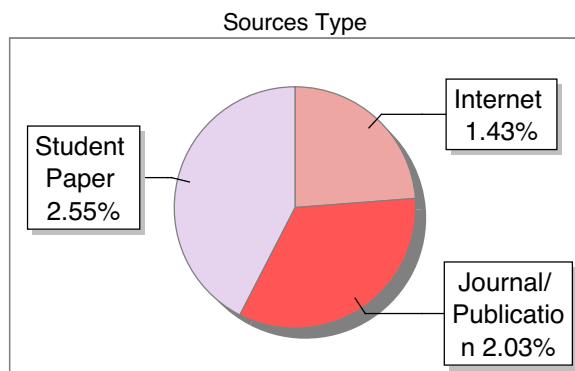
The Report is Generated by DrillBit Plagiarism Detection Software

### Submission Information

Author Name	sushmitha
Title	LearnSync
Paper/Submission ID	4144610
Submitted by	gururajs@sjec.ac.in
Submission Date	2025-07-29 15:25:36
Total Pages, Total Words	75, 11787
Document type	Project Work

### Result Information

Similarity **6 %**



### Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	<b>0 %</b>
Excluded Phrases	Not Excluded

### Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

