

PROJECT PROPOSAL

GROUP PROJECT

IIT271-2

Group No: IIT 05

SMART COMMERCE CORE

INDUSTRIAL INFORMATION TECHNOLOGY
DEGREE PROGRAM

Department of Computer Science and
Informatics

Faculty of Applied Sciences

Uva Wellassa University of Sri Lanka

2025

DECLARATION

We hereby declare that the project will be developed by us and will be our own effort and that no part will be plagiarized without citations under the supervision of Ms.K.A.A.Chathurangi, [ayesha.c@uwu.ac.lk]. This Project Proposal is submitted for the partial fulfillment of the requirement of the course unit IIT372-2, Project II for the industrial information technology degree program.

Group Details:

Group No: IIT 05

No.	Name of the Student	Index Number	E-mail address	Signature
1	K..Kavisana	UWU/IIT/22/090	iit22090@std.uwu.ac.lk	
2	A.H.M. Ayas	UWU/IIT/22/083	iit22083@std.uwu.ac.lk	
3	R.M.T.Madubhashini	UWU/IIT/22/081	iit22081@std.uwu.ac.lk	
4	A.G.P.P. Nandasiri	UWU/IIT/22/030	iit22030@std.uwu.ac.lk	

Supervisor Details:

Name of the Supervisor	E-mail	Contact Number
Ms.K.A.A.Chathurangi	ayesha.c@uwu.ac.lk	0774086621

Name of the Co-supervisor	E-mail	Contact Number
Ms. Y.M.A.P.S. Yapa	achiniyapa98@gmail.com	

Project Approval:

.....

Ms.K.A.A.Chathuragi

Date

Contents

1. Introduction.....	1
1.1 Project Title.....	1
1.2 project description	1
1.3 Project Background and Motivation.....	2
1.4. Problem in Brief	3
1.5. Proposed Solution.....	4
1.6 Project aims and objectives	4
1.6.1 Project Aim	4
1.6.2 Project Objectives	5
2. Methodology.....	6
2.1 Requirements Phase.....	6
2.2 Design Phase.....	7
2.3 Implementation Phase.....	8
2.4 Testing Phase	8
2.5 Deployment Phase	9
3. Requirements Identification.....	11
3.1 Functional Requirements	11
3.2 Non-functional Requirements.....	13
3.3 System requirements (Hardware / Software).....	15
3.3.1. Software	15
3.3.2. Hardware.....	17
3.4 User Roles.....	18
4. Project Plan.....	21
5. References.....	22

1. Introduction

1.1 Project Title

- SMART COMMERCE CORE

1.2 Project Description

Smart Commerce Core is a full-cycle Learning Management System tailored specifically for commerce-oriented teaching institutions. This robust system caters to academic, administrative, and financial operations, unifying principal stakeholders-students, tutors, parents, and administrators-under one umbrella of digital presence.

The system manages courses efficiently, logic-based auto-evaluation using the C language, and dynamically instantiated content with JavaScript, PHP, and HTML. Tutors upload study materials, quizzes, and assessments; students register for courses, download resources, and upload their assignments; parents view reports on student progress and other details, make payments; and administrators manage the whole application with role allotments to tutors and monitor usage of the system.

Featuring a MySQL database, grades, attendance, and revenues can be monitored in real time. Its predictive capabilities enable educational institutions in solving academic concerns before it becomes a serious issue. Integrated payment gateways and notice boards assist in financial management and communications.

Smart Commerce Core goes further in endorsing ease of use, performance insights, and effective engagements in learning, making itself the next-generation LMS solution for the real-world modern commerce academy.

1.3 Project Background and Motivation

With the rapidly changing education world, especially in commerce, there is an increasing call for digital transformation. Conventionally, academic matters involving examinations, student lists, attendance, and fees are cliché in their vocabulary; this is slow, error prone, and inefficient. Since many commerce academies still work with office-based/manual systems for admissions, marking, correspondence, etc., student engagement and academic performances at times can be negatively affected.

While the demand for individualized learning, real-time feedback, and seamless interaction among students, parents, tutors, and administrators is greater than ever, the degree of necessity of a centralized, smart solution has never been more. Yet, existing LMS usually cannot cater to the peculiarities of the workflows and account operations peculiar to commerce-oriented institutions and tuition centers.

The Smart Commerce Core - Smart Commerce Academy Portal fills this chasm. This project endeavors to design a role-based, integrated full-fledged academy management system pertinent to commerce education. For carrying out this much-needed automation of business processes, more efficient management of academic matters, and transparent communication among all stakeholders, the system is designed to simultaneously incorporate logic-based assessment modules in C, dynamic content delivery using HTML, PHP, and JavaScript, and strong data processing in MySQL.

Hence, while taking care of technical issues in education management, this solution unables.

1.4. Problem in Brief

The conventional management systems used by commerce academies face a bevy of issues that hinder both the teaching process and administrative efficacy. Manual methods may delay the giving of feedback to students and considerably increase the workload before tutors. This might result in a communication gap amongst tutors, students, and parents, leading to violations of deadlines, untracked student performance, and unwillingness of students to engage in the study.

There is usually a challenge for parents in being updated about a child's academic progress or renumeration in time due to non-availability of centralized real-time performance and fee reports. Attendance marking, giving academic support, and fee collection activities, among others, consume time when improperly managed and are error-prone.

Together, these inadequacies block the path of student success, foster demotivation among the tutors, and take a turnaround into the reputation of the institution itself. Hence, a central, automated, role-based platform is highly required which makes academic monitoring easier, administrative activities smoother, and opens better communication among all stakeholders.

1.5. Proposed Solution

Smart Commerce Core envisions a fully web-centric Learning Management System specially tailored for commerce schools and institutes. The system establishes the interface for the academic, communication, and financial functionality by way of dashboards of functions for Admin, Tutors, Students, and Parents.

Tutors can upload study materials, past lot, and model papers, make quizzes, and grade student performances through an auto-marking engine that is built with C logic. Students can handle course registrations, download resources, take quizzes, receive instant feedback, view their attendance, and submit assignments online. On the other hand, parents get to view the academic progress of their children, including attendance and grades, and pay fees securely via the integrated payment gateway.

Admins, however, are a higher level of supervisors for monitoring access, giving tutors access to subjects, publishing announcements on the portal, and also for generating reports in detail about student marks, attendance, and financial transactions. Running on a strong tech stack comprising HTML, JavaScript, PHP, C, and MySQL, the system guarantees responsive front-end design, impeccable logic as per requirements, secure data handling for the institutions, and real-time analytics for decision-making.

1.6 Project aims and objectives

1.6.1 Project Aim

The aim of this project is to design a centralized system of academy management for Smart Commerce Core that provides an easy-to-use digital platform for students, tutors, parents, and administrators. The system is designed to enhance the efficiency of academic and administrative tasks through an integration of courses, quizzes, performance monitoring, attendance, and fee payment in one place. It further increases parental involvement in the education of their children by giving instant access to the performance of scholarship and easy fee payment facilities.

1.6.2 Project Objectives

- **Make Easy Academic Access** -Offer students a simple-to-use online platform to access courses, download learning material instantly, and keep track of their academic progress simply.
- **Make Efficient Content & Assessment Management**-Make it easy for tutors to upload notes, previous papers, and quizzes, and also students' marks management via a simple interface.
- **Make Academic Operations Easy**-Make easy coordination among tutors, students, parents, and administrative staff simple to schedule classes, assessment, and academic reporting easily.
- **Make Communication Transparency Better**-Provide timely and relevant information to all stakeholders through real-time alerts, performance notices, and announcements.
- **Provide Accurate Performance Monitoring**-Enable timely grading of quizzes and assignments using automated grading technology, while providing instant access to results.
- **Support Institutional Development**-Encourage academic excellence and operational efficiency through electronic learning tools, automated fee processing, and wise performance intelligence, thereby boosting the reputation and reach of the institution.

2. Methodology

The SDLC model is being rigorously adopted by Smart Commerce Core implementation for a sequential approach and good delivery quality. Each phase is being executed conscientiously to tackle existing user demands and technical environments, and this discipline will also help Smart Commerce Core to attain future success and happiness.

2.1 Requirements Phase

The requirement gathering process involves identifying precise functional and non-functional requirements from administrators, tutors, students, and parents. Facts are currently being collected through:

- Interviews and questionnaires
- Analysis of the existing LMS system

The outcome of this phase is delivered in the form of a Software Requirements Specification (SRS) document, and this structured approach will continue to guide the system's development.

2.2 Design Phase

The system and user interfaces will be designed based on the collected requirements. The following will be the key design activities:

- **MySQL Database Schema Designing:** Structured relational data models will be created to efficiently store and manage system data.
- **PHP Backend Design Planning:** Designs will be prepared for server-side operations using PHP, including data processing, user authentication, and backend logic.
- **Wireframing for Role-Based Dashboard and UI/UX:** Wireframes will be developed for each user role Admin, Tutor, Student, and Parent to plan page layouts and navigation paths, ensuring a clear and smooth user experience.
- **HTML** will be considered during wireframing for structuring the page layouts and content hierarchy.
- **CSS** will be applied to ensure responsive styling, consistent theming, and visual appeal for all user roles, using stylesheets, Flexbox, Grid, and media queries.
- **JavaScript** will be used to make the UI interactive, enable real-time updates, and render user-specific content dynamically.
- **ER Diagrams and Flowcharts** will be created to illustrate entity relationships and data flow across various system modules.

2.3 Implementation Phase

- The development phase will fetch system components using a combination of technologies and a modular design approach.
- PHP will handle all server-side processing, including communication with the database, session handling, and execution of business logic.

2.4 Testing Phase

Step-by-step testing will be performed to ensure the system is bug-free and robust:

- Unit Testing will be conducted for individual modules.
- Integration Testing will be carried out to ensure proper interaction between modules.
- System Testing will be executed to validate the functionality of the complete platform.
- Bugs will be identified and resolved through an iterative process until stability is achieved.

2.5 Deployment Phase

The system and user interfaces will be designed based on the requirements collected earlier. The following will be the key design activities:

- **MySQL Database Schema Designing:** Structured relational data models will be created to efficiently store and manage system data.
- **PHP Backend Design Planning:** Designs will be prepared for server-side operations in PHP, including data processing, user authentication, and business logic.
- **Wireframing for Role-Based Dashboard and UI/UX:** Wireframes will be created for each role Admin, Tutor, Student, and Parent to plan page layouts and navigation paths. These wireframes will help ensure a clear and smooth user experience.
- **HTML** will be used during wireframing to define page structure and layout.
- **CSS** will be applied for responsive styling, visual consistency, and appealing theming across all roles, using stylesheets, Flexbox, Grid, and media queries.
- **JavaScript** will be utilized to make the UI interactive, enable real-time updates, and render user-specific content.
- **ER Diagrams and Flowcharts** will be developed to illustrate entity relationships and data flow across system modules.

3. Requirements Identification

3.1 Functional Requirements

1. User Management

- Admin approve, create, and manage user accounts.
- Admin assign tutors to subjects or classes.
- Role-based dashboards (Admin, Tutor, Student, Parent).

2. Course & Resource Management

- Tutors upload course content (PDF, notes, past papers).
- The tutors create quizzes with C logic-based marking.
- Students can register and view the material.
- Students can send in assignments from the website.

3. Automated Testing

- System auto-marks the quizzes based on the C logic-based marking logic. Instant feedback to students.
- Performance reports with weak/strong areas graphically displayed (Java Script/C).

4. Parent Monitoring Capabilities

- Parents see grades, attendance, and student progress.
- Parent payments or tuition.

5. Alerts and Notifications

- Parent and student notifications of impending due dates (assignment, quiz, fees).
- Tutor reminders to submit.

6. Financial Features

- Payment gateway (credit/debit card).
- Admin can see payment history and print invoices.

7. Analytics & Reports

- Real-time student dashboards of performance, attendance, and finance.
- Predictive analysis of performance based on analytics (Java Script/C).
- Quiz rankings and overall academic trends.

3.2 Non-functional Requirements

1. Performance

- The system will display real-time update of information without any noticeable lag.
- Quiz answers will be displayed within less than 2 seconds of submission

2. Scalability

- The system will be capable of concurrent access by thousands of users, including campuses, tutors, and parents.
- The system will support easy addition of new campuses or subjects.

2. Security

- The system will incorporate role-based access control for guaranteeing that sensitive information is protected.
- All personal and financial information will be encrypted.

3. Usability

- The system will have simple, intuitive interfaces for all users.
- It will have multilingual support where required to promote user convenience.

5. Availability

- The system will be available 24 hours a day, 7 days a week with an uptime of over 99%.
- It will have automatic backup of important data every 12 hours.

6. Maintainability

- The system will use maintainable coding methodologies using Java and C.
- It will have bug tracking and error logging functionalities.

7. Compatibility

- The system will be hosted by all major browsers (Edge, Firefox, Chrome) on an HTML/JavaScript-based frontend.
- It will feature a mobile-friendly interface for tablets and smartphones.

3.3 System requirements (Hardware / Software)

3.3.1. Software

Development Environment: Visual Studio Code will be utilized as the principal IDE for frontend (React) and backend (PHP).

Database Management System: MySQL will be the relational database system to handle student, user, and financial data.

Web Server: XAMPP will be employed to run a local server environment in order to test PHP-MySQL web applications.

Version Control: Version control and collaboration will be done using Git and GitHub.

Project Management: Trello or Asana will be used for task assignment, sprint planning, and monitoring.

Documentation: Google Docs and Google Drive will be used to write and synchronize documents such as user guides, design documents, and requirement specifications.

a. Backend: PHP

- Vanilla PHP will be employed for flexibility and direct access to the codebase, enabling tailored features for educational needs.
- Development time will be longer than in frameworks such as Laravel or Symfony but will provide better optimization for logic-specific features such as grading schemas.
- PHP will be natively integrated with MySQL for effective data operations.

b. Frontend: Frontend: JavaScript, HTML, CSS

- The frontend will be built using React to achieve responsiveness and role-specific UI experiences for students, tutors, parents, and admins.
- Its component-based architecture and virtual DOM will allow scalable and fast updates to the UI.

c. Database: MySQL

- MySQL will be used as a robust relational database system that is suitable for structured academic data such as grades, attendance, and finance.
- In-built PHP support for MySQL will provide secure and effective storage and retrieval of data.

d. Hosting /Infrastructure:

- The application shall be hosted on a cloud server using SSL encryption to allow secure access.
- A custom domain will be bought, and HTTPS will be enabled to offer secure communication.

3.3.2. Hardware

- **Development Devices:** Developers will code and test on laptops or desktops with at least 8GB RAM, multi-core processors, and SSD storage to make coding and testing effective.
- **Server Infrastructure:** The system will be deployed on cloud platforms such as AWS, Digital Ocean, or Firebase Hosting in order to ensure scalability and high availability.

3.4 User Roles

1. Administrator (Admin)

- Primary Responsibility: System setup and maintenance
- The admin will approve and manage student, tutor, and parent accounts.
- Subjects or classes shall be assigned to tutors.
- The admin will create courses and school calendars.
- They shall develop system analytics, reports, and revenue dashboards.
- Admin will manage the fee system. They will track payments.
- They will handle user roles and system platform updates.

2. Tutor

- Main Task: Provision of academic material and preparation of students
- The tutors will upload study material such as PDFs, videos, and notes.
- They will set up and manage quizzes based on C based logic.
- They will mark assignments and provide feedback on grades.
- Tutors will monitor the performance of students and identify weak spots.
- They will communicate with students via messages or discussion forums.

4. Student

- Primary Responsibility: Academic learning and participation
- Students will attend scheduled classes.
- They will view and download study material.
- They will submit assignments and attempt quizzes.
- Students will view their grades, attendance, and performance reports.
- They will see submission deadlines and notifications.

4. Parent / Guardian

- Primary Responsibility: Monitoring academic performance and payment
- Parents/guardians will monitor their child's grades, attendance, and performance.
- They will track quiz attempts and assignment submissions.
- They will pay course fees securely via an online portal.
- They will receive necessary academic reminders and notifications.

User Role Access Matrix

Feature	Student	Tutor	Parent	Admin
View Courses	✓	✓	✓	✓
Enroll in Courses	✓			✓
Submit Assignment	✓			
Download Study Materials	✓	✓		✓
Upload Study Materials		✓		✓
Design Quizzes		✓		✓
Attempt Quizzes	✓			
View Grades	✓	✓	✓	✓
View Attendance	✓	✓	✓	✓
Automated Grading		✓		✓
Predictive Performance Reports		✓	✓	✓
Fee Payment	✓		✓	✓
Manage User Accounts				✓
Register/Login	✓	✓	✓	✓
Configure System Settings				✓

✓ Indicates the user role has permission for the corresponding feature.

4. Project Plan

Deployment and Operations: Cloud server installation, web hosting server installation, ability of monitoring services to install quickly, securely, and reliably.

[illegible]

5.References

- MySQL Documentation. (n.d.). MySQL 8.0 Reference Manual. Retrieved from [\[https://dev.mysql.com/doc\]](https://dev.mysql.com/doc)(<https://dev.mysql.com/doc>)For understanding database commands and optimization.
- GitHub Docs. (n.d.). Introduction to Git and Version Control. Retrieved from [\[https://docs.github.com\]](https://docs.github.com)(<https://docs.github.com>)Referenced for managing source code collaboratively.
- W3Schools – HTML, CSS, Java, C++, MySQL: <https://www.w3schools.com>
- MySQL Official Documentation – <https://dev.mysql.com/doc/>
- JavaTpoint – Full-stack development tutorials

