

Tribhuvan University

Faculty of Humanities and Social Sciences

“AcademiaPlus”-Academic Management System

A PROJECT REPORT

Submitted to

Department of Computer Application

Lumbini City College

*In partial fulfillment of the requirements for the Bachelors in Computer Application*

Submitted by

Suman Khatri

(T.U.Reg.No:-6-2-1134-60-2020)

Chitra Bahadur Thapa

(T.U.Reg.No:-6-2-1134-42-2020)

Under the Supervision of

Dinesh Neupane

# **Chapter: 1 Introduction**

## **1.1 Introduction**

In today's educational environment, the management of academic activities such as attendance, scheduling, and grading is often fragmented and inefficient. This project seeks to develop an integrated solution to address these challenges. The proposed AMS will provide a unified platform to facilitate real-time attendance tracking, dynamic routine scheduling, and efficient marks management, thereby improving administrative efficiency and academic outcomes.

## **1.2 Problem Statement**

1. Authentication and Authorization: The AMS must provide secure login and registration mechanisms to ensure that only authorized users can access the system.
2. Real-time Data Management: The system should allow for real-time updating and retrieval of attendance, routine schedules, and marks.
3. Privacy and Security: Ensuring the privacy and security of user data is crucial to gain user trust.

## Performance: The AMS should handle high loads without compromising performance.

## **1.3 Objective**

1. User-Friendly Interface: Provide an easy-to-navigate interface for seamless user experience, ensuring that users can easily access and use the functionalities of AcademiaPlus.
2. Privacy and Security: Ensure that users feel safe and their personal information is protected while using AcademiaPlus.
3. Reliability: Ensure the application runs smoothly without crashing, even during times of high usage.
4. Multimedia Support: Enable users to share various types of media, including images, videos, documents, and voice messages, to enrich the academic management experience.

# **Chapter: 2 Methodology**

## **2.1 Requirement Identification**

1. Study of Existing System:

The study of existing academic management systems such as Blackboard, Moodle, and Canvas reveals several strengths and areas for improvement. These platforms offer features like attendance tracking, schedule management, and grade management, which are essential for educational institutions. However, issues such as user interface complexity, limited customization, and varying levels of integration capabilities are prevalent. While some platforms excel in providing comprehensive features and user support, others may fall short in offering a seamless user experience and efficient performance. By analyzing these aspects, we can identify best practices to adopt and the gaps to address in AcademiaPlus to ensure a superior user experience and enhanced functionality.

1. Literature Review:

The literature review for the development of AcademiaPlus involves examining various scholarly articles, industry reports, and technical documents to gather insights on current technologies and best practices. Key findings indicate that existing academic management systems often face issues related to user interface complexity, integration capabilities, and scalability. Studies highlight the importance of modern front-end frameworks like Next.js for dynamic user interfaces, robust back-end technologies like Node.js and Express for scalability, and secure databases like PostgreSQL for data integrity and security. Additionally, the review underscores the necessity of user-centric design principles and efficient workflow automation to enhance user experience. These insights provide a solid foundation for developing AcademiaPlus, addressing these critical areas effectively.

1. Requirement Analysis:

The requirement analysis for AcademiaPlus involves gathering detailed information on user needs and preferences through various methods such as interviews, surveys, and focus group discussions. By understanding what features users find essential, such as two-way attendance tracking, routine management, and marks management, we can define the core functionalities that AcademiaPlus must have. This process ensures that the final product will meet user expectations and address any existing shortcomings in current academic management systems. The analysis also emphasizes the importance of an intuitive interface, robust performance, and seamless integration with other educational tools to provide a comprehensive solution for academic institutions.

## **2.2 Feasibility Study**

1. Technical Feasibility:

We'll evaluate the hardware, software, and skilled personnel needed, as well as potential technical challenges and solutions. This includes ensuring the chosen technologies (Next.js, Tailwind, TypeScript, Express, Node.js, PostgreSQL) can support a scalable and robust system.

1. Operational Feasibility:

We'll assess how well the new system integrates with current workflows and processes. This includes evaluating ease of use and training needs for teachers, students, and administrators to ensure smooth adoption.

1. Economic Feasibility:

We'll perform a cost-benefit analysis to determine the financial viability. This includes estimating development and operational costs, as well as the expected return on investment (ROI) to ensure the project is economically sustainable.

## **High-Level Design of System**

1. Methodology of the Proposed System:

Using the Waterfall methodology for developing AcademiaPlus involves a linear and sequential approach to project development. Here’s how Waterfall principles can be applied to the development of AcademiaPlus:

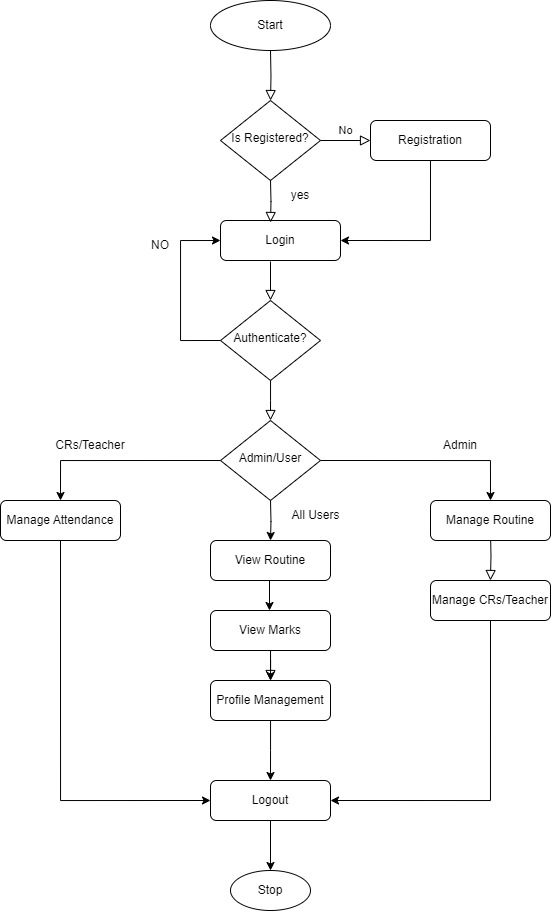
* Requirements Analysis: We start by gathering all the necessary information about what AcademiaPlus should do. This involves talking to teachers, students, and administrators to understand their needs. We document all the requirements, such as two-way attendance tracking, routine management, and marks management.
* System Design: Once we have a clear understanding of the requirements, we design the system architecture. This includes creating detailed specifications for the software and hardware components, as well as designing the database schema. We also create wireframes and mockups to visualize the user interface and plan the user experience.
* Implementation:In this phase, we begin the actual coding of AcademiaPlus. We follow the design specifications and develop each module one by one. For example, we start with the attendance tracking module, followed by routine management, and then marks management. We use Next.js, Tailwind, TypeScript, Express, Node.js, and PostgreSQL to build the system.
* Integration and Testing:After the individual modules are developed, we integrate them into a complete system. We conduct thorough testing to ensure that each module works correctly and that the entire system functions as expected. This includes unit testing, integration testing, and system testing to identify and fix any issues.
* Deployment:Once the system has been thoroughly tested and all issues have been resolved, we deploy AcademiaPlus to the production environment. We set up the necessary servers and ensure that the system is accessible to users. We also perform final checks to confirm that everything is working correctly.
* Maintenance: After deployment, we provide ongoing support and maintenance for AcademiaPlus. This includes monitoring the system for any issues, providing updates and patches, and making improvements based on user feedback. We also ensure that the system remains secure and performs optimally.



**Figure 1 Waterfall Methodology**

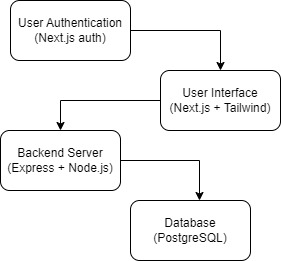
1. Flowchart

The flowchart illustrates the steps of a user interacting with AcademiaPlus. Users begin by logging in or registering if they don't have an account. After logging in, the main dashboard is displayed, showcasing various features and options. Class representatives (CRs) and one designated teacher can manage attendance, ensuring accurate tracking of student participation. All users can view the routine to stay updated on class schedules, but only admins have the authority to update it. Users can also manage and view marks, allowing for efficient academic tracking and performance evaluation. Additionally, users can access their profiles to update personal information and settings. Notifications keep users informed about important updates and events. After completing their tasks, users log out, ending the process.



**Figure 2 Flowchart**

1. Working Mechanism of Proposed System:



**Figure 3 Working Mechanism of System**

1. Description of Algorithms:

* User Authentication:

Step 1: User Input: The user enters their email and password on the login page.

Step 2: Backend Auth Request: The front-end application (built with Next.js) sends the credentials to the Express.js backend.

Step 3: Validation: The backend validates the credentials against the PostgreSQL database. If valid, the backend returns a user token. If invalid, an error message is returned.

Step 4: User Session: The front-end application stores the user token in local storage or context for session management. Upon successful authentication, the user is redirected to the main dashboard.

* Two-Way Attendance Management:

Step 1: Mark Attendance: Class representatives (CRs) or the designated teacher marks attendance for a specific class using the front-end application.

Step 2: Send Attendance Data: The front-end sends the attendance data to the backend.

Step 3: Store Attendance Data: The backend stores the attendance data in the PostgreSQL database.

Step 4: Update Attendance Status: The backend updates the attendance status and notifies the front-end application of the changes.

Step 5: View Attendance: Users can view their attendance records. The front-end fetches the attendance data from the backend and displays it.

* Routine Management:

Step 1: View Routine: All users can view the class routine. The front-end requests the routine data from the backend.

Step 2: Fetch Routine Data: The backend retrieves the routine data from the PostgreSQL database and sends it to the front-end.

Step 3: Display Routine: The front-end displays the routine in a user-friendly format.

Step 4: Update Routine (Admin Only): Admins can update the routine. The front-end sends the updated routine data to the backend.

Step 5: Store Updated Routine: The backend stores the updated routine data in the PostgreSQL database and notifies users of the changes.

* Marks Management:

Step 1: Enter Marks: Teachers enter students' marks using the front-end application.

Step 2: Send Marks Data: The front-end sends the marks data to the backend.

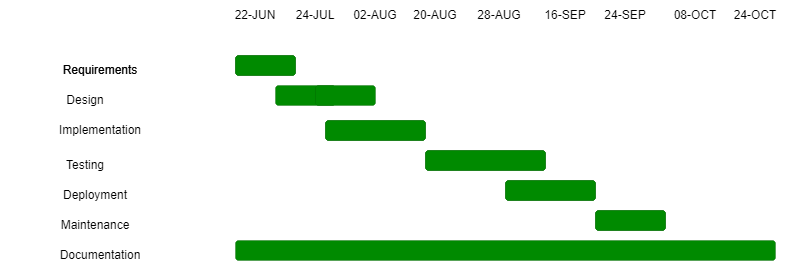
Step 3: Store Marks Data: The backend stores the marks data in the PostgreSQL database.

Step 4: View Marks: Students and teachers can view the marks. The front-end fetches the marks data from the backend and displays it.

Step 5: Update Marks (Teachers Only): Teachers can update marks if needed. The front-end sends the updated marks data to the backend, which stores the changes in the PostgreSQL database and updates the front-end.

# **Chapter: 3 Gantt Chart**

A Gantt chart will be created to outline the project timeline, including key milestones and deliverables. This will provide a clear roadmap for the project and help in tracking progress.



# 

**Figure 4 Gantt Chart**

# **Chapter: 4 Expected Outcome**

The expected outcome of AcademiaPlus is a secure, efficient academic management system that streamlines user authentication, two-way attendance tracking, routine management, and marks management. It will provide an intuitive, responsive interface, ensuring a seamless user experience across devices. The platform will offer reliable data storage and retrieval, support for updates by admins, and easy access to records for students and teachers. Additionally, robust security measures will protect user data and privacy, making the system compliant with data protection regulations. The system will also facilitate communication between students, teachers, and administrators, enhancing collaboration and transparency within the institution. Overall, AcademiaPlus aims to deliver a scalable, high-performance solution that meets the diverse needs of modern educational institutions, improving efficiency and overall academic management.

# **References**

|  |  |
| --- | --- |
| [1] | A. Jain, “Enhancing Educational Administration and Academic Management: A Study of the School Management System,” *International Scientific Journal of Engineering and Management,* May 2024. |
| [2] | R. West and R. S. Davies, “Technology Integration in Schools,” January 2014. |
| [3] | S. A. Pramono, M. Hanif and K. Kraugusteeliana , “Management and Utilization Strategies of Educational Management Information Systems to Enhance Students' Academic Performance,” *Indo-MathEdu Intellectuals Journal,* May 2024. |
| [4] | D. .. D. .. K. M. M. Felisiya, “Exploring the Influence of Learning Management System Constructs on Student Satisfaction and Academic Performance.,” *European Economics Letters,* February 2024. |