IJOBTECH
INTERNATIONAL JOURNAL OF SMART BUSINESS AND TECHNOLOGY

Vol. 1 No. 1 (2025)

Web-Based Student Monitoring Application Development Using a System Design Approach

Muhammad Naufal Ardiansyah [1], Hendrico Edhent Surya Pratama [2], Muhammad Fahrur Rozi [3], Farrel Zikri Suryahadi [4], Mohammad Zahri Valent Affandi [5]

[1,2,3,4,5] *Informatika, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur , Jl Raya Rungkut Madya, Gunung Anyar, Surabaya and 60294, Indonesia*

Abstract

This study aims to design and develop a web-based student monitoring application under a Design System approach. This application is offered as a solution to problems in monitoring student academic activity, i.e., time delay in information, non-centralized data, and limited real-time visualization of study progress. This system is developed with ReactJS, ExpressJS, and Node.js technology supported by containerization with Docker and traffic management with Nginx. To make the development and update process automatic and efficient, CI/CD pipeline is used with GitHub Actions. The Design System approach allows the formation of a consistent, structured, and easy-to-maintain user interface, as well as improving readability and user experience in the long term. This application is equipped with some main features, i.e., role-based login (admin, lecturer, student), academic progress dashboard, task notification system, time management suggestion, and API integration for direct academic data synchronization. Pilot testing was conducted by involving students as well as lecturers as users of the system and concluded that this application was able to facilitate the effectiveness of academic communication and make monitoring of student performance more convenient. This system was also proven to be responsive across different devices as well as easy to use. With its flexibility and scalability, this application is worthy of being implemented widely in higher education institutions in Indonesia.

Keywords: Web-Based; Student Monitoring; Design System; UI/UX; Academic Communication

Article History:

Received:	*25 August 2025*
Revised:	*13 October 2025*
Accepted:	*24 October 2025*
Published:	*31 October 2025*

1- Main Text

1-1- Introduction

In the current digital era, educational institutions are increasingly required to adopt information technology to improve the efficiency and effectiveness of academic management. The advancement of information technology has significantly transformed various sectors, including higher education, which is now shifting toward integrated and efficient digital systems [1], [2]. One of the key challenges faced by universities is how to monitor student activities in a real-time, structured, and adaptive manner to user needs, particularly in support of academic guidance and administrative decision-making [3].

The student monitoring process encompasses multiple aspects, including attendance, assignment submission, and academic progress. However, academic monitoring systems that are still manual or fragmented often hinder both universities and students from obtaining accurate and up-to-date information. These inefficiencies in academic data tracking and the lack of real-time reporting tools can disrupt communication between students and advisors, resulting in poor academic oversight.

To address these issues, this study presents the design and development of a web-based student monitoring application. The system leverages modern technologies such as ReactJS for the frontend, ExpressJS for the backend, and a microservices-based architecture containerized using Docker. It is also equipped with secure authentication mechanisms, automated notifications, and a time management feature to help students achieve academic success more effectively.

[1] **CONTACT**: 23081010090@student.upnjatim.ac.id

1-2- Methodology

1-2-1- Type of Research

This is a Research and Development (R&D) type of study aimed at producing a product in the form of a web-based student monitoring application. The focus is on designing, developing, and testing an academic information system accessible online that supports centralized and real-time student activity.

1-2-2- System Development Method

An iterative software engineering method was adopted, which includes the following stages:

a. Requirements Analysis

Identifying user needs (students, academic advisors, and admins) through observation and literature study.

b. System Design

Designing system architecture, database structure, authentication flow, and UI/UX using tools such as architectural diagrams, ERDs, and wireframes.

c. System Implementation

- Technologies used:
- Frontend: ReactJS
- Backend: ExpressJS (Node.js)
- Database: MongoDB/PostgreSQL
- Deployment: Docker & Docker Compose
- Reverse Proxy: Nginx
- CI/CD: GitHub Actions

d. System Testing

Using Black Box Testing and User Testing to verify functionality.

e. Evaluation and Improvement

Based on feedback and testing results, improvements are made for further development cycles.

1-2-3- Data Collection Techniques

Data collection in this study is conducted through the following methods:

a. Literature Study: Reviewing academic references related to academic information systems and modern web development technologies.

b. Observation: Observing user needs in an academic campus setting.

c. Interviews and Questionnaires: Conducted with students and lecturers to gather insights and evaluate the system.

1-2-4- Testing Method

Testing is conducted in two main phases:

a. Black Box Testing

This method is used to test major system features such as login, dashboard, notifications, profile updates, and logout, to ensure that the system responds correctly to each type of user input.

b. User Testing

The application is tested by 13 users (10 students and 3 lecturers) based on predefined usage scenarios. Observations and user feedback are recorded to evaluate the interface and system functionality.

1-2-5- Development Tools and Environment

Programming Language	: JavaScript (ReactJS & Node.js)
Frameworks	: ExpressJS, ReactJS

Database	: PostgreSQL
Container Platform	: Docker
Web Server	: Nginx
Code Repository and CI/CD	: GitHub & GitHub Actions
Testing Device	: Laptop (Windows/Linux)

1-2-6- Activity Diagram

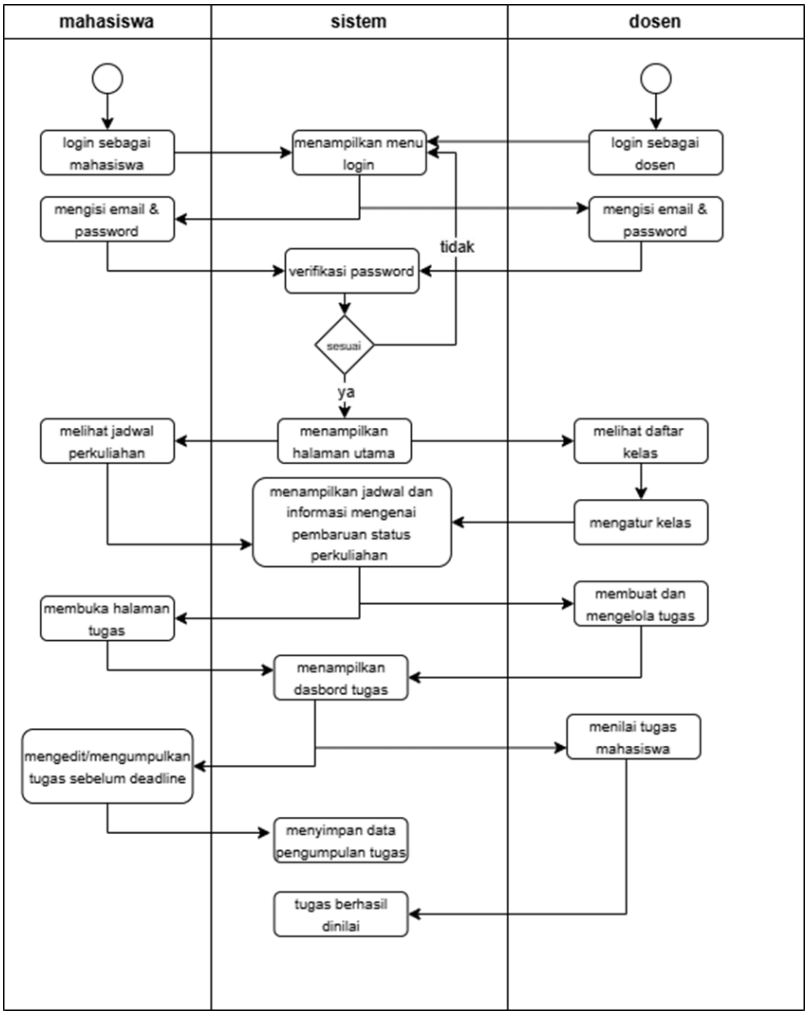

Figure 1. Activity diagram.

Student

a. Login as Student

The process begins when the student accesses the login page and enters their email and password credentials.

b. Enter Email & Password

The system receives the login input and proceeds to verify the credentials.

c. If login is successful, the student is granted access to the system, where they can:

- View their class schedule
- Open the assignment page
- Edit or submit assignments before the deadline

System

a. Display Login Page

The system initially displays the login interface for all users.

b. Verify Password

The system checks whether the provided email and password match existing data.

c. If the credentials are invalid, the process is terminated and the user remains on the login page.

d. If the credentials are valid, the system will:

- Display the main/home page
- Show the student's class schedule and academic status
- Display the assignments dashboard
- Save submitted assignment data
- Notify the student that the assignment has been successfully graded

Lecturer

a. Login as Lecturer

The login process is the same as for students. After successful login, the lecturer is directed to their dashboard.

b. After logging in, the lecturer can:

- View the list of their classes
- Manage class data
- Create and manage student assignments
- Evaluate and grade student submissions

1-3- Functional Features and Outcomes

1-3-1- Landing Page

The initial interface of the web-based student monitoring application serves as a crucial element that forms the user's first impression while also functioning as the main gateway to the system. This page is designed with a modern, minimalist, and responsive approach, balancing both aesthetic and functional aspects. The primary focus is to create an intuitive interface that is easy to understand and quickly accessible by all user categories—students, academic advisors, and institutional administrators.

Visually, the landing page displays the institution's logo or the application's name at the top as a system identifier. The main feature highlighted on this page is the login form, which explicitly provides options to sign in as a student, advisor, or admin. This role-based separation ensures that the authentication process aligns with the user's context and grants access to relevant features according to their access rights.

The UI design of the landing page utilizes components from a previously developed Design System, including:

- A neutral and high-contrast background color to maintain visual comfort,
- Consistent typography using a modern sans-serif font for readability across various screen sizes,
- Buttons and input forms with adequate padding, clear focus indicators, and inline input validation,
- High responsiveness to screen sizes, ensuring optimal display on both desktop and mobile devices.

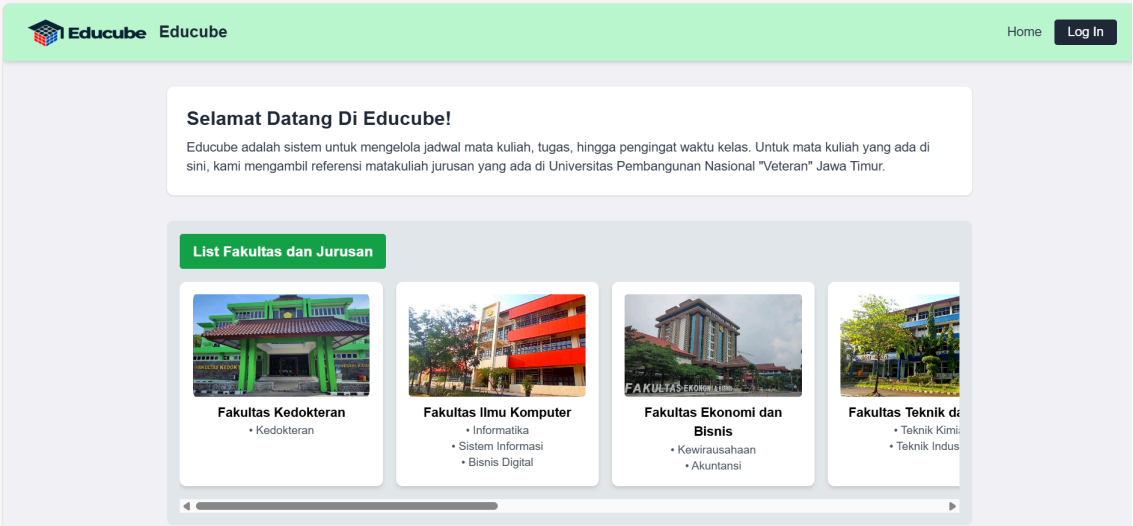

Figure 2. Landing page.

1-3-2- Login Page as College Student

Authentication is a critical aspect in the development of web-based information systems, especially those involving personal data and students' academic activities. Therefore, the login authentication process in this student monitoring application is designed with a focus on security, reliability, and user convenience.

The dedicated student login page allows users to access the system by entering their Student Identification Number (NIM) and password. The NIM serves as a unique identifier within the academic system, while the password protects user privacy and prevents unauthorized access. The login form is designed with a user-friendly approach, incorporating real-time client-side validation to immediately notify users of input errors before submission to the server.

From a system architecture perspective, the login authentication is handled by a backend built with Express.js, which receives login data from a ReactJS frontend. Upon receiving a login request, the server performs the following steps:

- **Input validation**: Ensuring that the NIM and password formats meet the required standards.
- **Data matching**: Searching for user records in the database based on the NIM.
- **Password verification**: Using hashing algorithms (e.g., bcrypt) to compare the entered password with the encrypted one stored in the database.
- **Authentication token generation**: If verification is successful, the server generates a JSON Web Token (JWT) or manages a session and returns it to the client.
- **Secure session management**: The token/session is securely stored in the browser, for example using an httpOnly cookie to prevent XSS attacks.

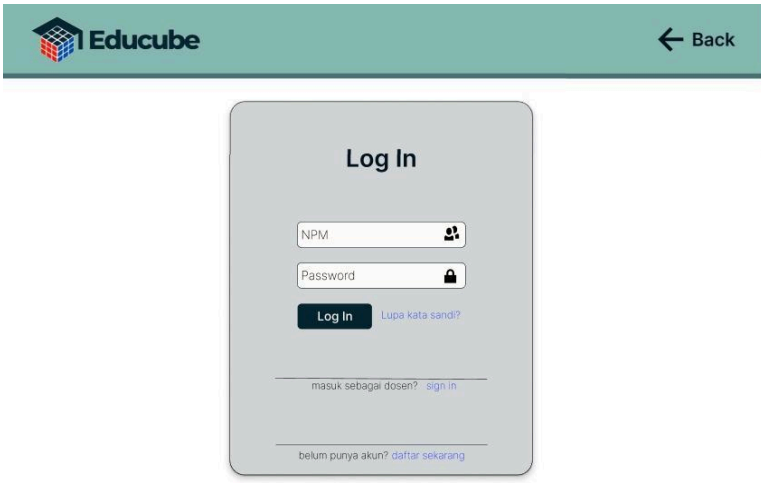

Figure 2. Login page college student.

1-3-3- Main Menu

The main menu serves as the core navigation element in this web-based student monitoring application. It functions as the central control point for users—particularly students—to access all available system features quickly, efficiently, and in a well-organized manner. The design of the main menu is guided by usability principles and design consistency, which are key components of the Design System implemented in the application's interface development.

Visually, the main menu is presented as a navigation sidebar positioned on the left side of the screen. This sidebar features icons and text labels representing each available feature, making them easily identifiable even for first-time users. On smaller screens (such as mobile devices), the sidebar transforms into a collapsible hamburger menu to optimize screen space.

The main components of this menu include:

- **Dashboard**

The primary page that summarizes key student activities, including attendance records, active assignments, and academic progress.

- **Classes**

This feature allows students to view their enrolled courses along with related information such as schedules, instructors, and assignment lists for each subject.

- **Student Profile**

Provides access for students to view and update personal information such as full name, email address, phone number, and academic program.

- **Notifications**

Contains a list of system-generated alerts, including assignment deadlines, schedule changes, and messages from academic advisors.

- **Time Management**

A dedicated feature offering time management recommendations based on the number of tasks, class schedules, and the student's activity patterns. This helps students plan their academic workload more effectively.

- **Logout**

This button allows users to securely end their session and remove the authentication token or session data from the client side.

Each menu item is designed to be responsive and dynamic, where the menu display adjusts based on the role of the logged-in user (student, lecturer, or admin). In this initial development phase, the focus is placed on the student role, with a menu structure that is easily extendable to support other user roles in the future.

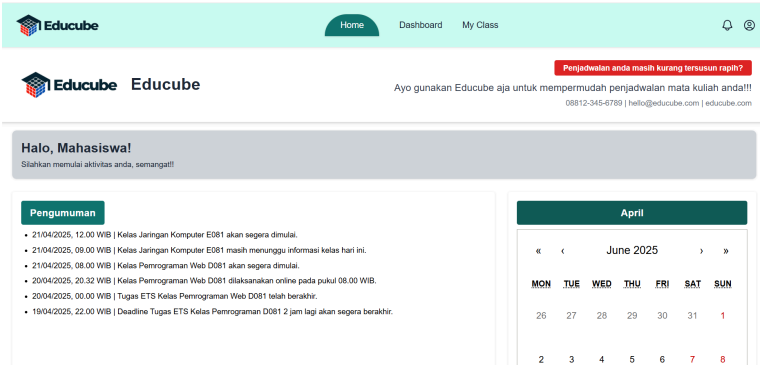

Figure 3. Main menu.

1-3-4- Dashboard

The dashboard serves as the central information hub in this web-based student monitoring application. Its primary function is to provide a concise, real-time, and structured overview of the student's academic status. Designed to be the first page users see after a successful login, the dashboard plays a strategic role in facilitating efficient interaction between students and the system.

Overall, the dashboard layout consists of several info cards and simple charts, organized in a responsive, modular structure. The key components of the dashboard include:

- **Attendance Summary**

Displays the student's attendance statistics as percentages and session counts. If attendance falls below the threshold, the system highlights it with a visual warning indicator (e.g., red/orange colors).

- **Active Assignments**

Summarizes all pending or approaching-deadline assignments, including course names, deadlines, and completion status.

- **Study Progress**

Visually represents academic progress using elements such as progress bars or donut charts, reflecting completed tasks, average grades, and achievements per semester.

- **Recent Notifications**

Lists announcements from instructors or system-generated alerts, including class updates, additional materials, or activity reminders.

- **Time Management Recommendations**

When available, the system provides personalized time management suggestions based on the student's schedule and workload, helping them prioritize academic tasks more effectively.

1-3-4-1- Purpose and Function of the Dashboard

The dashboard is developed to present the most relevant information needed by students in their daily academic activities. Its objectives include:

- Providing a quick overview of attendance status, assignments, and course progress.
- Delivering important notifications directly to ensure students don't miss deadlines or schedule changes.
- Enhancing student awareness and responsibility toward their academic activities through easily understandable visualizations.

1-3-4-2- Dashboard Information Components

The dashboard consists of several information cards and simple charts, organized in a modular and responsive structure. The main components include:

- **Attendance Summary**

Displays the student's attendance statistics in percentage and session count. If attendance falls below the threshold, the system shows a visual warning indicator (in red or orange).

- **Active Assignments List**

Summarizes all pending or near-deadline assignments, complete with the related course, due date, and completion status.

- **Study Progress**

Presents academic progress visually using elements like progress bars or donut charts, reflecting the number of completed tasks, average grades, and achievements per semester.

- **Recent Notifications**

Shows a list of announcements sent by lecturers or the system, such as class updates, additional material info, or event reminders.

- **Time Management Recommendations**

When available, the system offers suggestions for time management based on the student's schedule and workload to help prioritize study activities.

1-3-4-3- Technology and Architecture

The dashboard components are built using the React JS framework, leveraging state management to dynamically display updated data without reloading the page. Data communication is handled through REST APIs from the Express.js backend, which retrieves the latest information from the database.

All displayed data is fetched via authenticated API requests, ensuring that only students with valid access tokens can view their information. The system supports real-time updates through periodic polling or, optionally, WebSocket technology for instant notifications.

1-3-4-4- Interface Design and Accessibility

The dashboard adopts a responsive design approach, allowing seamless access across various devices, both desktop and mobile. UI elements such as color, icons, typography, and layout follow the established Design System to maintain visual consistency and ease of navigation.

Visual indicators are designed to be intuitive:

- Green for good/complete status
- Orange/yellow for medium-level attention
- Red for critical status or actions required

In addition, the system ensures the dashboard is accessible to users with special needs by using sufficient color contrast and HTML structures compatible with screen readers.

1-3-4-5- Implementation Impact

Based on trial results and field observations, the dashboard has proven to be highly beneficial for students in managing their academic activities. Most respondents reported that the dashboard helped them become more disciplined and motivated to complete assignments and attend classes on time.

The dashboard not only displays data but also functions as a tool for self-reflection and personal decision-making. It supports a more focused, data-driven learning experience that fosters individual self-awareness.

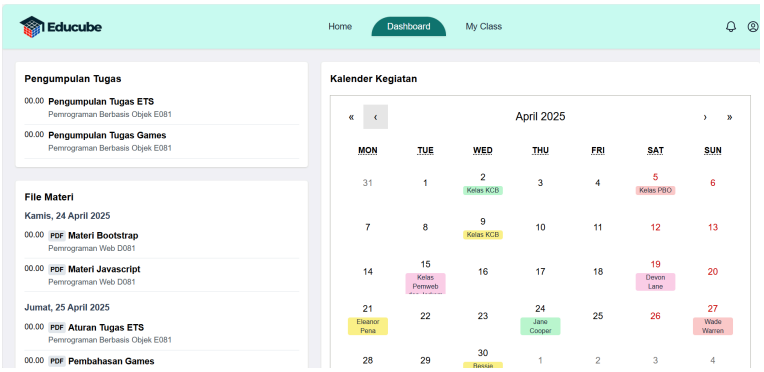

Figure 4. Dashboard.

1-3-5- College Student Class Design

1-3-5-1- Student Class View

The Student Class View feature functions to present all essential information related to the courses a student is enrolled in during a given semester. The displayed information includes the course name, instructor, class schedule, assignment list, attendance history, learning materials, and lecturer comments or notes.

Each class is displayed in the form of clickable cards or lists that allow students to view detailed information. Students can also access features such as uploading assignments, reading course materials, or communicating with lecturers through a discussion forum (if available).

The interface is built using ReactJS, and data is retrieved from an ExpressJS backend via REST APIs protected by authentication tokens. The UI design follows a consistent and responsive design system principle to ensure optimal usability across all devices.

This feature significantly helps students monitor their progress in each course independently, avoid assignment delays, and improve more structured academic communication.

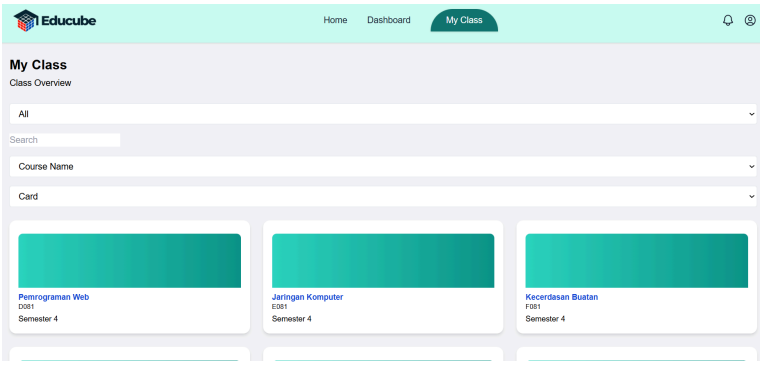

Figure 5. **Student Class View.**

1-3-5-2- College Student Profile

The **Mahasiswa Profile** feature allows users to view and update personal information such as name, email, and profile photo. Certain details, such as Student Identification Number (NIM) and study program, are set as read-only to maintain the accuracy of academic records.

The profile interface is designed to be responsive and user-friendly, built using ReactJS and connected to the ExpressJS backend through secure APIs. All data modifications are protected by a token-based authentication system and are validated before being saved.

This feature helps mahasiswa manage their academic identity independently, improve data accuracy, and reduce the administrative burden on campus staff.

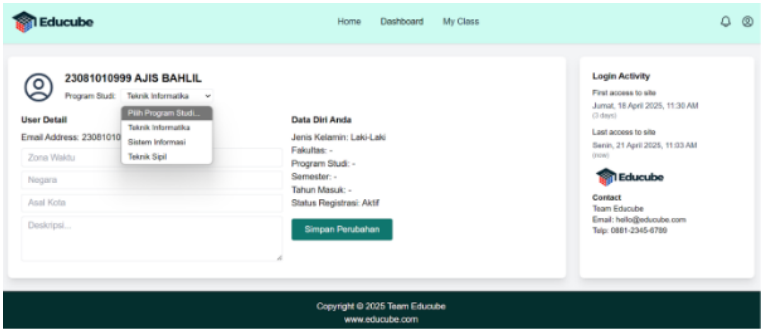

Figure 6. **College Student Profile.**

1-3-6- Notification

The **Assignment Notification** feature plays a crucial role in helping students monitor and manage their academic tasks in a timely manner. The system automatically displays notifications based on the assignment status and the time remaining until the deadline.

Notifications are presented in the form of pop-ups, numerical badges, or lists within a dedicated notification menu. The information displayed includes:

- Assignment title
- Associated course
- Deadline date
- Work status (not started, in progress, or completed)

The system logic is designed so that notifications appear progressively, for example:

- **Three days before the deadline** – an initial reminder is triggered
- **One day before the deadline** – a high-priority alert is shown
- **Overdue assignments** – marked with a special color indicator (e.g., red)

This feature is directly connected to the backend via an API that periodically checks student assignment data. Through this approach, students are encouraged to be more disciplined, avoid missing deadlines, and develop better academic task prioritization habits.

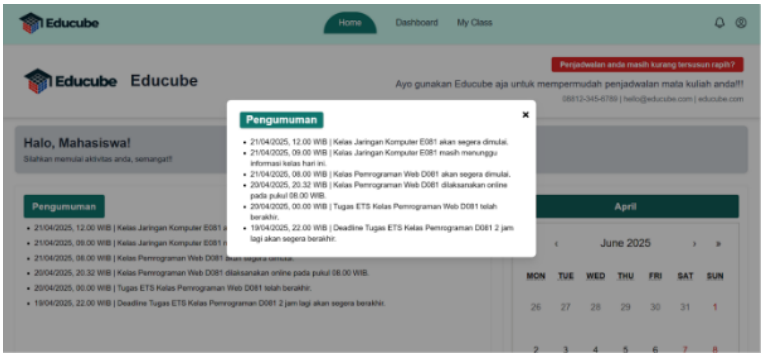

Figure 7. Notification.

1-3-7- Time Management Feature

The **Time Management Recommendation** feature is designed to help students manage their time effectively based on their academic activity data. This system provides personalized suggestions for study schedules, task completion, and prioritization. Recommendations are generated based on the following factors:

- Number of active assignments
- Assignment deadlines
- Weekly class schedule
- History of delays or past activity patterns

The system processes this data using a simple rule-based approach and generates suggestions in the form of daily or weekly schedules. These schedules are displayed in a calendar or list view and are dynamically updated whenever data changes—such as when a new assignment is added or a task is marked as completed. The interface is designed to be user-friendly, featuring color indicators and time labels such as *"Ideal Time to Work on Assignments," "Free Time,"* or *"Deadline Risk Period."*

This feature not only helps students avoid delays but also trains them to develop more structured and disciplined work habits, which are essential for long-term academic success.

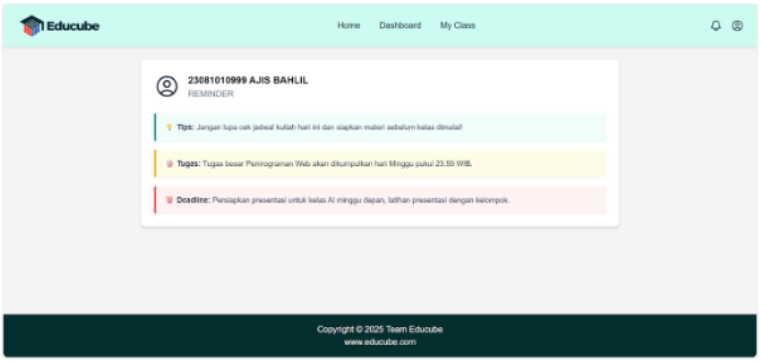

Figure 8. Time management feature.

1-3-8- Logout

The **Logout** feature functions to securely end the user session and prevent unauthorized access to the system after a student has finished using the application. Logout is a critical component of access control and security in web-based applications, particularly in academic environments that involve sensitive data. When the user clicks the "Logout" button, the system will:

- Remove the authentication token (such as JWT) from the browser's local storage or cookies.
- Redirect the user back to the login page or initial landing screen.
- Terminate the active session on the backend to prevent token reuse (in the case of session-based authentication).

The logout button is consistently placed in the main menu (either in the sidebar or profile dropdown), with a clear icon and label to ensure it is easily accessible. By providing a secure and user-friendly logout feature, the application safeguards students' personal data and encourages users to maintain account security—especially when using public or shared devices.

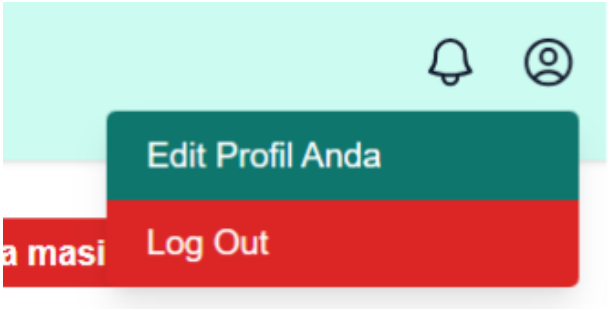

Figure 9. Logout.

1-4- Testing Results

1-4-1- User Testing

After completing internal system testing, the next phase was user testing, aimed at evaluating how the application functions in real scenarios and how users respond to the interface and features. This testing focuses on usability, comfort, and system effectiveness from the user's point of view.

1-4-2- Test Subject

The test involved 13 participants:

- 10 active students from various study programs, representing the primary users of the system
- 3 academic supervisors (lecturers) as academic stakeholders

1-4-3- Testing Procedure

Each participant was asked to use the application in a real-world scenario, performing the following tasks:

- Log in as a student
- Explore the dashboard and task information
- Access class details
- Edit or update profile data
- Respond to task notifications
- Follow time management suggestions
- Log out of the system

The testing was conducted over 1–2 days, with direct observation and post-usage satisfaction questionnaires.

1-4-4- Observations and User Feedback

- **90% of students** stated that the user interface is easy to understand and use.
- **80% of students** found the application helpful in organizing their task schedules and tracking academic progress.
- One lecturer suggested adding a feature to automatically generate a PDF summary of student activities for academic guidance evaluation.

Students also gave positive feedback on:

- Attractive visual design
- Intuitive navigation
- Ease of accessing assignment data

Some students suggested integrating the system directly into the campus academic system, stating it is more user-friendly than previous platforms.

1-4-5- Functionality Testing Results

Table 1. Testing Results Table.

Feature Tested	Result	Notes
Student Login	Success	Invalid inputs are rejected; successful login redirects to dashboard
Menu Navigation	Success	All menus are accessible and switch quickly
Dashboard	Success	Displays real-time data and accurate visualizations
Student Profile	Success	Data updates saved successfully; input validations function properly
Task Notifications	Success	Notifications appear on time and match task status
Time Management Recommendations	Success	Schedule appears according to user activity data
Logout	Success	Token is cleared; user is redirected to login screen

2. Declarations

2-1- Author Contributions

Conceptualization, Farrel Zikri Suryahadi and Hendrico Edhent Surya Pratama; methodology, Muhammad Fahrur Rozi; software, Muhammad Fahrur Rozi; validation, Mohammad Zahri Valent Affandi, Muhammad Naufal Ardiansyah and Muhammad Fahrur Rozi; formal analysis, Mohammad Zahri Valent Affandi; investigation, Hendrico Edhent Surya Pratama; resources, Muhammad Naufal Ardiansyah; data curation, Muhammad Naufal Ardiansyah; writing—original draft preparation, Mohammad Zahri Valent Affandi; writing—review and editing, Farrel Zikri Suryahadi; visualization, Muhammad Naufal Ardiansyah, Hendrico Edhent Surya Pratama, Muhammad Fahrur Rozi, Mohammad Zahri Valent, and Farrel Zikri Suryahadi; supervision, Muhammad Fahrur Rozi; project administration, Hendrico Edhent Surya Pratama; funding acquisition, Farrel Zikri Suryahadi. All authors have read and agreed to the published version of the manuscript.

2-2- Data Availability Statement

The data presented in this study are openly available to the public on GitHub at the following link: https://github.com/rozi8/website-pemberitahuan-akademis.

2-3- Funding

Funding information is not available. This research and system development were carried out independently by the authors without financial support from any external institution, grant, or funding agency.

2-4- Acknowledgements

The authors would like to express their sincere gratitude to the Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, for the invaluable support provided during the entire development and testing process of this system. The faculty’s contribution, including access to laboratory facilities, guidance from academic staff, and a conducive research environment, played a crucial role in enabling the team to implement and refine the proposed solution. Their continuous encouragement and availability of infrastructure significantly enhanced the quality and efficiency of the system’s development. This research would not have been possible without their dedication and collaborative spirit in fostering innovation and academic growth.

2-5- Institutional Review Board Statement

Ethical review and approval were waived for this study due to the nature of the research, which did not involve medical, psychological, or biological experiments on human subjects.

2-6- Informed Consent Statement

Informed consent was obtained from all subjects involved in the study, including students and faculty participants in the system user testing.

2-7- Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. All ethical considerations including plagiarism, informed consent, misconduct, data integrity, and redundant publication have been thoroughly observed.

3- References

[1] Abelia, L., Gunawan, D., & Kurniawan, B. (2024). *Perancangan Sistem Monitoring Akademik Mahasiswa Berbasis Web*. Jurnal Teknologi Informasi dan Pendidikan, 7(1), 25–34.

[2] Junaedy A. H. et al. (2023). *Transformasi Digital Perguruan Tinggi: Tantangan dan Strategi*. Jurnal Sistem Informasi, 11(4), 121–132.

[3] Lulu, I., Santosa, T., & Rachmadani, F. (2022). *Monitoring Kehadiran dan Nilai Siswa Melalui Aplikasi Web dan Mobile*. Jurnal Teknologi Pendidikan, 6(2), 45–56.

[4] Pradana, G. A. (2024). *Implementasi Teknologi Digital dalam Pendidikan Tinggi*. Jurnal Inovasi Teknologi Pendidikan, 8(1), 1–10.

[5] Reinhard, Y. A., Nugroho, S., & Andika, M. (2024). *User-Centered Design dalam Pengembangan Aplikasi Pendidikan*. Jurnal UX dan Interaksi Digital, 3(1), 22–35.

[6] Siringoringo, H., & Alfaridzi, M. N. (2024). *Efektivitas Penggunaan Platform Pembelajaran Digital di Lingkungan Universitas*. Jurnal Teknologi dan Pendidikan, 4(2), 88–97.