Hoang Ngoc

COMPREHENSIVE STUDENT MANAGEMENT

2025

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Hoang Ngoc
Title	Comprehensive Student Management
Year	2025
Language	English
Pages	53
Name of Supervisor	Harri Lehtinen

The primary objective of this research is to develop a comprehensive student management web application that enables teachers to efficiently manage students remotely. This system is designed to be accessible from any location and platform, helping educators save time while streamlining administrative tasks.

The system was built upon online learning management principles, allowing teachers to oversee student progress, provide support, and facilitate communication without the need for physical presence in a classroom. Several studies, such as E-learning and Digital Education Management, have explored the digital transformation in education and the role of management systems in enhancing teaching effectiveness. Based on these foundations, this system aims to help teachers track student performance, assess behavior and ethics, and facilitate seamless communication between students and educators.

The application is developed using JavaScript (Node.js) with Express.js as the backend framework, and MySQL for data storage. The frontend was built with HTML, CSS, and Vanilla JavaScript (JS), ensuring a responsive and user-friendly interface. Once fully developed, this system will be accessible to teachers and students across all educational levels, making it a versatile tool for modern educational institutions.

Keywords JavaScript, Nodejs, HTML, CSS, MySQL, Express.js

CONTENTS

FIGURES

TABLES

ABBREVIATIONS

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
HTTP	HyperText Transfer Protocol
SQL	Structured Query Language
UML	Unified Modeling Language
MVC	Model-View-Controller
UX	User Experience
UI	User Interface
RDBMS	Relational Database Management System

1 INTRODUCTION

Currently, student management systems in Vietnam predominantly rely on traditional approaches, with records being managed and stored manually on paper. This method is not only time-consuming for teachers but also increases the likelihood of document loss or misplacement. Moreover, students often need to be physically present in class to discuss coursework or address concerns directly with their teachers, which limits their time for self-study and research.

While many studies have focused on developing e-solutions for monitoring student performance (e-learning and digital education management systems) (Bates, 2015,p 418-419; Pappas, 2015), most of these systems have their own limitations. Factors such as limited communication features, minimal capabilities for student performance measurement, and poor user experience have hindered their adoption as effective e-learning resources in educational contexts (Nicol & Macfarlane-Dick, 2006).

To address these problems, this thesis presents a Comprehensive Student Management System, designed to provide a simple mechanism for interaction between students and lecturers. It reduces reliance on in-person meetings, while also enabling teachers to actively manage grades and assess students. Furthermore, students can view their academic results and receive teacher feedback online in real time, fostering continuous improvement in their learning experience.

The system is built with a user-friendly interface to ensure ease of use and accessibility for both educators and learners. It also simplifies routine administrative tasks by enabling the simultaneous management of multiple students across various classes and courses, offering significant organizational benefits to educational institutions.

1.1 Use of AI in This Thesis

In this thesis, I used ChatGPT to generate ideas and refine my own thoughts to create a more comprehensive and well-structured content. I have utilized AI to enhance the fluency and coherence of my writing while ensuring that the original message and intent remained intact. Additionally, I have used Grammarly to check and improve grammatical accuracy.

I am committed to maintaining the authenticity and integrity of the content while respecting copyright regulations. Before incorporating any external information, I have carefully verified its source. If the information already exists, I provide proper citations to ensure accuracy and adherence to ethical research practices.

2 SCOPE OF THE APPLICATION

This thesis application functions more as an academic support website that accommodates all types of students, spanning a variety of academic levels from middle school to college. It provides a comprehensive toolkit of applications to assist in managing students and courses. Key features include course management, student record tracking, grading, student assessments.These functions are integrated into a single platform, reducing the need for multiple applications and streamlining usability and communication.

The platform also allows both students and teachers to receive important notifications from the school. This ensures they stay informed about institutional news and announcements.By consolidating these activities into a single, user-friendly system, this solution enhances: organization, communication, overall educational efficiency.

2.1 Research Objective

The objective of the thesis is to create a management system for student and pupils using web-based on platform offers fast proccessing, secure data storage, and efficient management. These systems facilitate administrative processes, reduce manual errors, and provide instant access to information, helping students achieve a better educational experience. They provide a secure and dependable solution for today’s educational institutions by centralizing information and automating processes.

To prevent the loss of time and resources in storing and searching for information, we propose a system where teachers can simply log in and instantly access student data, class divisions. Additionally, teachers can stay updated with school notifications and quickly share important announcements with students.

This not only streamlines the educational process, fostering a more connected and cohesive academic environment, but also contributes to a faster, more efficient, and responsive learning experience.

2.2 Contributing factors

The application to be developed enables schools to better manage student and course processes—drastically reducing errors and increasing data reliability.It reduces the burden of manual management efforts by automating and centralizing processes. This, in turn, lowers the risk of human error (HelioCampus, 2017, p9).

Easier retrieval and storage of data support high-volume access. This also means that information can be quickly retrieved and updated as needed, enabling better decision-making in educational contexts (Siemens & Long, 2011).

With more accurate data and fewer administrative tasks, the technology allows teachers to focus on teaching—it's as simple as that. This efficiency, combined with the scalability of the system, enables educational institutions to handle an ever-growing amount of data without compromising the quality of the educational process (Nguyen & Nguyen, 2023).

2.3 Intended Organization

This website has been specifically designed for educational institutions, administrators, teachers, and students. It serves as a unified tool that supports the management of educational processes by organizing courses, tracking student performance, and enabling smooth communication between teachers and learners. Specifically, the website represents a comprehensive, user-friendly response to the evolving demands of modern education.

By incorporating all of these features into a single interface, the system not only increases efficiency but also helps schools maintain tighter monitoring and coordination.Whether it is school administrators overseeing operations, teachers monitoring classes, or students logging in to learn, the platform provides a streamlined solution for all users.

2.4 Technology Used in the Thesis

The implementation of this web application uses a state-of-the-art technology stack to ensure efficiency, user-friendliness, and scalability. For the frontend, HTML and CSS were used to provide a well-structured, visually appealing, and user-friendly interface designed to optimize the user experience. HTML defines the structure and layout of the web pages, while CSS is responsible for styling the components, managing layout, and ensuring compatibility across a wide range of devices.

JavaScript was used on the backend, along with Express.js, a minimalist yet powerful web framework for Node.js. Express.js simplifies the development of server-side logic and the management of HTTP requests, routing, middleware, and other features, enabling the backend to perform efficiently and reliably.

For data storage, MySQL was used as the relational database management system (RDBMS). It serves as the central location for storing, retrieving, and managing data across various categories such as users, courses, students, scores, and communication logs.By integrating these technologies, the application provides both teachers and students with secure, real-time access to data, consistent performance, and reliable functionality.

3 THEORETICAL FRAMEWORK BASED ON SOURCES

This research builds upon pioneering studies in the field of applying information technology to educational administration. One major reference is Prieto et al. (2011), *"Educational Data Mining: An Overview."* In this research, the author explores the use of data mining methods to analyze and enhance various aspects of teaching and learning, contributing to the improvement of education quality through technology.

Another foundational work is by Davis (1989), which presents a theoretical model explaining how users—teachers, administrators, and students—perceive and accept new technological tools in educational settings. This theory is essential for understanding how such systems can be integrated and how they influence user satisfaction and system usability.

Additionally, this work is closely aligned with Garrison & Anderson (2003), *"E-Learning in the 21st Century: A Community of Inquiry Framework for Research and Practice."* This framework emphasizes the importance of collaborative learning environments and teacher-student interaction through the use of technology. It helps assess how IT solutions can serve not only as tools for optimizing administrative processes but also as enablers of effective learning through improved communication and engagement.

3.1 Software Development Processes

Agile is a methodology designed to manage smaller, incremental product developments while improving collaboration between customers and developers. It emphasizes transparency throughout every part of the process to ensure clarity not only for developers but also for customers.

With Agile, projects are divided into smaller iterations or *sprints*, delivering value incrementally and allowing for ongoing feedback. At the end of each sprint, a review is conducted where teams gather feedback and organically adapt their processes based on that input.

Developers working individually can also reflect on their performance and refine their process in future iterations. This practice encourages personal growth and supports the iterative improvement that Agile promotes.

The Agile methodology fosters adaptability, enables fast responses, and prioritizes customer satisfaction. It ensures that teams and clients remain informed, involved, and integrated throughout the software development process (Omonijie, 2024).

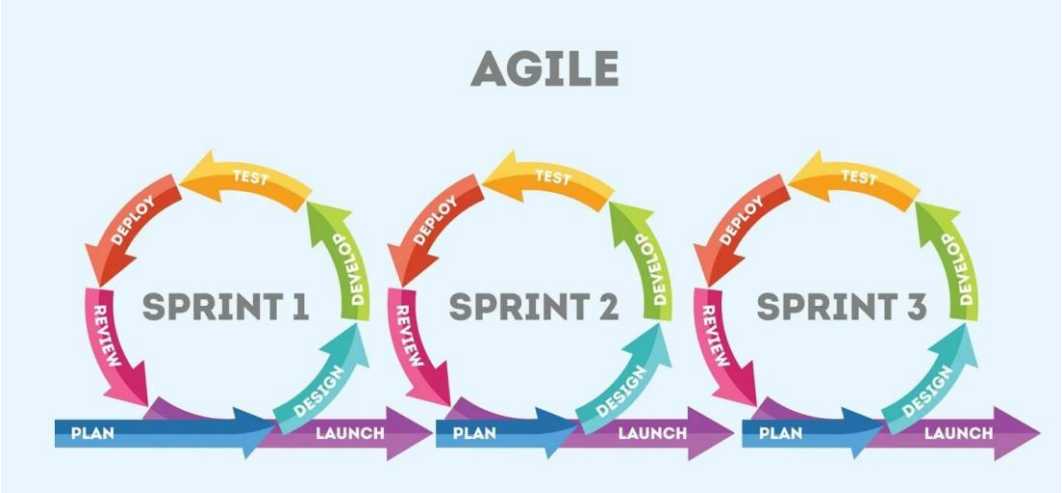


Figure 1. Agile Software Development Process (Slawek-Polczynska, A. 2020).

3.2 Requirements and Design Methods

UML (Unified Modeling Language) is a standard language used for specifying, visualizing, constructing, and documenting the artifacts of software systems. It is particularly effective for analyzing and visualizing systems by representing them through multiple types of diagrams. These diagrams help

both technical experts and end-users develop a clear understanding of the system's structure and behavior before actual development begins.

Developers and teams benefit from using UML by gaining a clearer understanding of the system they are going to build, helping them stay on track and avoid confusion during later stages of development. UML also reduces the risk of errors, as it presents the system design in a clear and organized manner, leading to improved communication among team members.

UML provides a standardized way to "write down" system elements so that everyone involved—assuming they are familiar with UML—can interpret them consistently. The use of UML in system design is crucial to the software development cycle and plays a key role in effectively communicating design concepts across teams (Booch, Rumbaugh, & Jacobson, 2005, pp. 1–45).

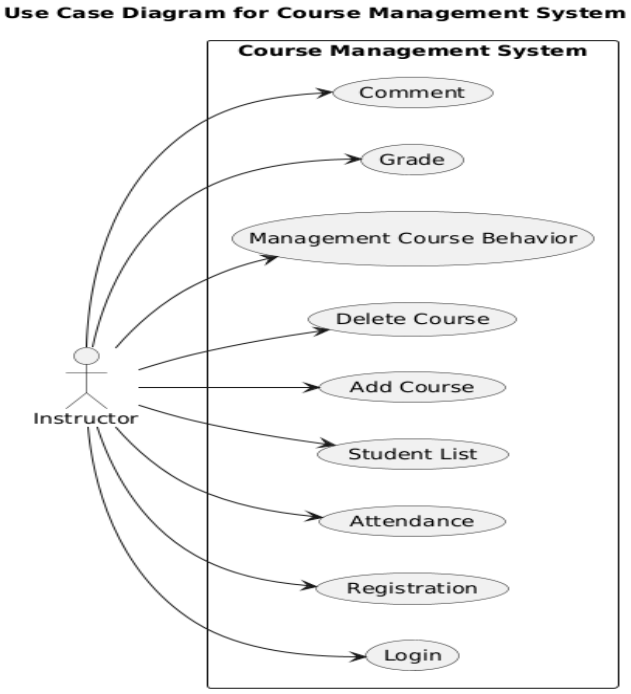


Figure 2. Use case diagram.

3.3 Architectural Models

The Model–View–Controller (MVC) is a standard software design pattern used in desktop and web application development. It divides the program into three parts: the Model contains data and logic; the View is what users see and interact with on the screen; and the Controller connects the Model and View, determining how they communicate (Wikipedia, 2025). These components manage the discrete actions of the application, making the code easier to understand, maintain, and scale (MDN Web Docs, n.d.).

This architecture is popular and widely used for complex systems with various features such as student management, course management, and notifications (MDN Web Docs, n.d.).

The Model is responsible for storing data and application state. It manages information about students, courses, and results while maintaining data integrity and structure (MDN Web Docs, n.d., para. 3).

The Views are responsible for displaying the user interface, including the student list, scores, attendance, and reminders. They aim to visualize data clearly and understandably (Model-view-controller, n.d).

The Controller acts as the intermediary between the View and the Model. It handles user input, updates the Model accordingly, and then refreshes the View to reflect these changes, enabling user interaction (Sheldon, 2023).

In this project, we use the MVC pattern to separate concerns. For example, when a teacher adds or drops a course, the Controller responds to the action, interacts with the Model to process the data, and then instructs the View to update the user interface accordingly.

4 PLANNING AND IMPLEMENTATION

Test scores can be entered and saved by teachers, and students can view their scores in real time once the system confirms them. Meanwhile, students can also receive regular assessments of their behavior and ethics from teachers throughout the course to support their improvement. This platform facilitates smooth communication and exchange of feedback between students and teachers as needed.

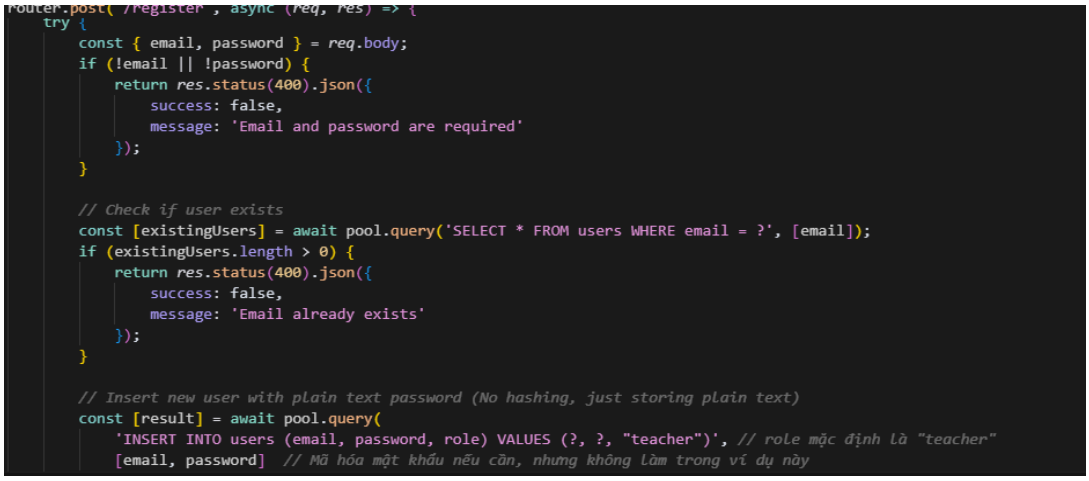
4.1 Implementation Code

4.1.1 Registration

The registration code requires an email and password. It then checks if the email already exists; if it does, a notification is sent informing the user that the email is already registered. Otherwise, the registration is successful.

Modify	user_id	email	password	role	name
edit	1	1@gmail.com	1	student	*NULL*
edit	2	2@gmail.com	2	teacher	*NULL*
edit	3	3@gmail.com	3	student	*NULL*
edit	23	default_email@example.com	default_password	student	B

Figure 3. User's registration.

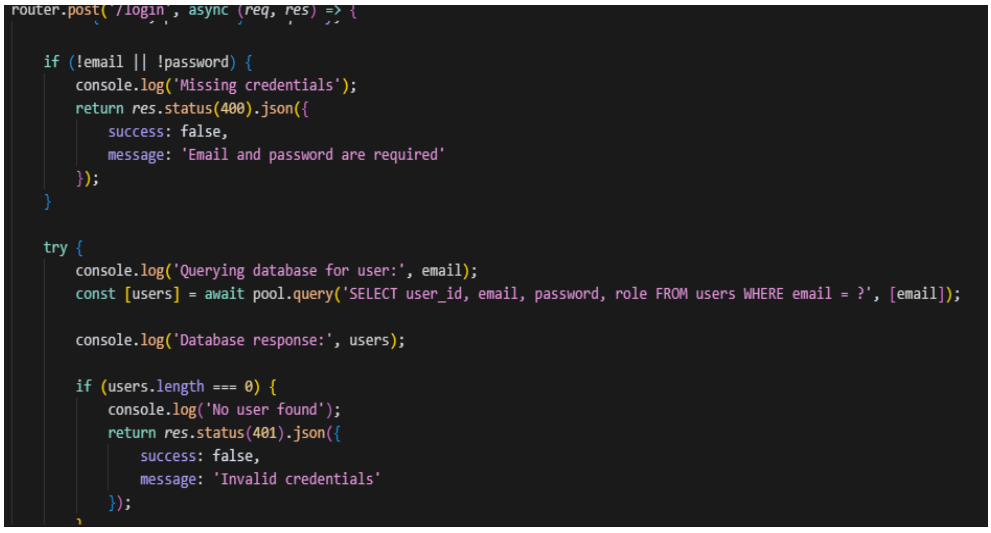


```
router.post('/register', async (req, res) => {
    try {
        const { email, password } = req.body;
        if (!email || !password) {
            return res.status(400).json({
                success: false,
                message: 'Email and password are required'
            });
        }

        // Check if user exists
        const [existingUsers] = await pool.query('SELECT * FROM users WHERE email = ?', [email]);
        if (existingUsers.length > 0) {
            return res.status(400).json({
                success: false,
                message: 'Email already exists'
            });
        }

        // Insert new user with plain text password (No hashing, just storing plain text)
        const [result] = await pool.query(
            'INSERT INTO users (email, password, role) VALUES (?, ?, "teacher")', // role mặc định là "teacher"
            [email, password]  // Mã hóa mật khẩu nếu cần, nhưng không làm trong ví dụ này
```

Figure 4. Registration code.

4.1.2 Log In

When attempting to log in, users must enter both their email and password. If either field is missing, the system recognizes the incomplete credentials and notifies the user about the missing information. Once all required fields are filled, the system checks the database to verify whether the credentials are correct and then returns the result.



```
router.post('/login', async (req, res) => {

    if (!email || !password) {
        console.log('Missing credentials');
        return res.status(400).json({
            success: false,
            message: 'Email and password are required'
        });
    }

    try {
        console.log('Querying database for user:', email);
        const [users] = await pool.query('SELECT user_id, email, password, role FROM users WHERE email = ?', [email]);

        console.log('Database response:', users);

        if (users.length === 0) {
            console.log('No user found');
            return res.status(401).json({
                success: false,
                message: 'Invalid credentials'
            });
```

Figure 5. Log in code.

4.1.3 Courses

The courses code retrieve all courses from the database to display to users. Users can then add a new course by providing its name and ID or delete a course by name and ID. For all actions, users receive notifications indicating whether the operation was successful or not.

```
// GET - Lấy tất cả khóa học cùng giáo viên
router.get('/', async (req, res) => {
  try {
    const [results] = await db.query('SELECT * FROM courses');
    res.json(results);
  } catch (err) {
    console.error('Error fetching courses:', err);
    res.status(500).json({
      success: false,
      message: 'Error fetching courses',
      error: err.message,
      sqlError: err.sqlMessage
    });
  }
```

Figure 6. Courses code.

Modify	course_id	name	created_at
edit	1B	Biology	2025-04-16 21:54:34
edit	1E	English	2025-04-25 21:31:29
edit	1F	Finnish	2025-04-24 22:56:47
edit	1H	History	2025-04-16 21:54:07
edit	1M	Math	2025-04-16 22:02:52
edit	1P	Physic	2025-04-16 22:04:57

Figure 7. Course's database.

4.1.4 Course Student Controller

The cousre student controller code display all students enrolled in a course and allow management of their grades and attendance. Teachers can add or remove students and manage other course-related functions. The system processes these commands, verifies them, and executes the appropriate actions to return the correct response.

```
router.get('/students/:course_id', async (req, res) => {
    const { course_id } = req.params;
    console.log('Fetching students for course:', course_id);

    try {
        // First verify the course exists
        const [courseRows] = await db.query('SELECT course_id FROM courses WHERE course_id = ?', [course_id]);
        if (!courseRows || courseRows.length === 0) {
            return res.status(404).json({
                success: false,
                message: 'Course not found'
            });
        }
```

Figure 8. Get all students in a course.

Modify	student_id	name	email	phone	created_at
edit	21	M	1@gmail.com	1	2025-04-24 23:59:58
edit	22	A	5@gmail.com	5	2025-04-24 23:54:53
edit	23	B	4@gmail.com	12345	2025-04-21 00:06:56

Figure 9. Students in database.

```
// Record attendance
router.post('/attendance', async (req, res) => {
    const { student_id, course_id, date, status } = req.body;
    console.log('Recording attendance:', { student_id, course_id, date, status });

    if (!student_id || !course_id || !date || !status) {
        return res.status(400).json({
            success: false,
            message: 'All fields are required'
        });
```

Figure 10. Record attendance.

Modify	id	student_id	enrollment_id	date	status
edit	1	23	2	2025-04-25	present
edit	2	23	2	2025-04-22	present
edit	3	23	3	2025-04-25	present
edit	4	23	3	2025-04-23	present

Figure 11. Attendance's database.

```
// Record grade
router.post('/grade', async (req, res) => {
    const { student_id, course_id, grade, comment, behavior } = req.body;
    console.log('Recording grade:', { student_id, course_id, grade, comment, behavior });
```

Figure 12. Record grade.

```
// Remove student from course
router.delete('/students/:course_id/enrollments/:student_id', async (req, res) => {
    const { course_id, student_id } = req.params;
    console.log('Removing student from course:', { student_id, course_id });
```

Figure 13. Remove student from course.

Modify	id	student_id	course_id	grade	comment	behavior	updated_at
edit	1	23	1H	10.00	good	excellent	2025-04-21 00:46:32
edit	5	23	1F	8.00	great	excellent	2025-04-25 22:11:36

Figure 14. Enrollment's database.

```
// Add student to course
router.post('/students/:course_id/enrollments', async (req, res) => {
    const { course_id } = req.params;
    const { student_id, name, email, phone } = req.body;
    console.log('Adding student to course:', { course_id, student_id, name, email, phone });
```

Figure 15. Add student to course.

4.1.5 Dashboard code

Dashboard code retrieve information from the database and display it on the dashboard in real time. The system sorts the data by age, allowing users to see the oldest items listed on the dashboard page.

```
const notificationList = document.getElementById('notificationList');
if (notifications.length > 0) {
  notifications.forEach(notification => {
    const listItem = document.createElement('li');
    listItem.textContent = notification.message;
    notificationList.appendChild(listItem);
  });
} else {
  const noNotificationsMessage = document.createElement('li');
  noNotificationsMessage.textContent = 'No new notifications';
  notificationList.appendChild(noNotificationsMessage);
}
catch (error) {
console.error('Error loading data:', error);
alert('Error loading data. Please try again later.');
```

Figure 16. Dashboard code.

4.1.6 Students Details Code

Students details code display the student's name, email, ID, and phone number to show that the student is enrolled in a course. The student can create comments, record behavior, or add scores. The system saves this information to the database and reloads it whenever accessed.

```
// Get student details
router2.get('/:id', (req, res) => {
    const id = req.params.id;
    db2.query('SELECT * FROM students WHERE id = ?', [id], (err, results) => {
        if (err) return res.status(500).json({ message: 'Database error', error: err });
        res.json(results[0]);
    });
});

// Update student feedback
router2.post('/:id/feedback', (req, res) => {
    const id = req.params.id;
    const { score, comment, behavior } = req.body;
    const query = 'UPDATE students SET score = ?, comment = ?, behavior = ? WHERE id = ?';
    db2.query(query, [score || null, comment || null, behavior || null, id], (err) => {
        if (err) return res.status(500).json({ message: 'Update failed', error: err });
        res.json({ message: 'Student info updated' });
    });
});
```

Figure 17. Student details.

4.2 Software Testing

4.2.1 Unit Testing

Unit testing is a level of software testing where individual units or components of software are tested separately. Since the tested units are generally small and have limited behavior, the testing process is easier to implement, organize, and document. It also allows for quicker analysis of test results, making it simpler to focus on specific targets and fix faults instead of searching through the entire system (Jorgensen, 2013, p. 58)

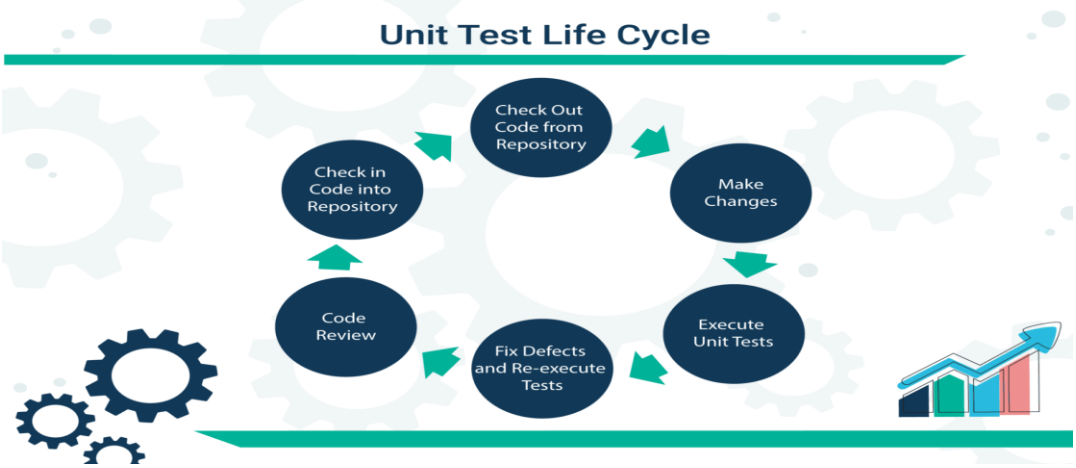


Figure 18. Unit Test: What is, Adventages and How works? (Codebox, 2022).

Unit testing is an effective method for testing features that are self-contained and do not rely on other modules. In a school management system, unit tests are performed on smaller components such as login, registration, course creation, student enrollment, attendance entry, grade input, and student evaluation. Each of these functions can be tested independently to ensure they work according to the system's requirements (Myers, Sandler, & Badgett, 2011, p. 107).

By testing each function as it is developed, developers can more easily identify the root cause of errors, which simplifies the debugging process and increases confidence in the application. Unit testing also facilitates faster updates and maintenance, as bugs can be fixed at the individual unit level without affecting the rest of the system. Thus, unit testing plays a central role in improving code quality and system stability, especially in modular systems like those used in student course management systems (Burnstein, 2003, p. 91).

4.2.2 Integration Testing

Integration testing involves testing the interfaces between multiple assembled modules that have already been verified through individual unit tests. Unlike unit testing, which targets single units, integration testing focuses on the interactions between integrated modules to ensure they function cohesively as an application. This includes checking the interactions, data flowand control functions of components to guarantee smoothcollaboration (Myers, Sandler, & Badgett, 2011, p. 135).

Figure 19. Integration test (Anderson, 2023).

For this task, integration testing is very important to verify how the different modules in the project interact with each other, such as testing the functionality of login and dashboard access, as well as marking attendance and saving results in the system. For example, the system should properly redirect the user to the dashboard upon successful login, where personalized content and features are presented according to the user's role. This assumes that the "separation of concerns" between authentication logic and user interface components always works as intended (Jorgensen, 2013, p. 84).

Additionally, this type of integration testing ensures that when a teacher records attendance, the data is processed and stored correctly in the backend. It verifies the correctness of the attendance module's logic and the accuracy of data flow between the controller and model in the MVC design. Such testing is crucial in systems based on the MVC architecture to ensure that the components (Model, View, and Controller) interact properly, handling requests, data updates, and user interfaces effectively (Burnstein, 2003, p. 115).

4.2.3 Acceptance Testing

Acceptance testing is a validation activity conducted to determine whether the system meets its technical and business requirements. The emphasis is on testing the behavior of the software in its operational environment and determining whether the system satisfies the customer's requirements (Myers et al., 2011, p. 281).

This type of testing is especially important in the late stages of development or at the end of each sprint in Agile, where feedback is received directly

from real users (i.e., students and teachers). This process provides confidence that the features released match the intended functionality and valid usage context (Crispin & Gregory, 2009, p. 143).

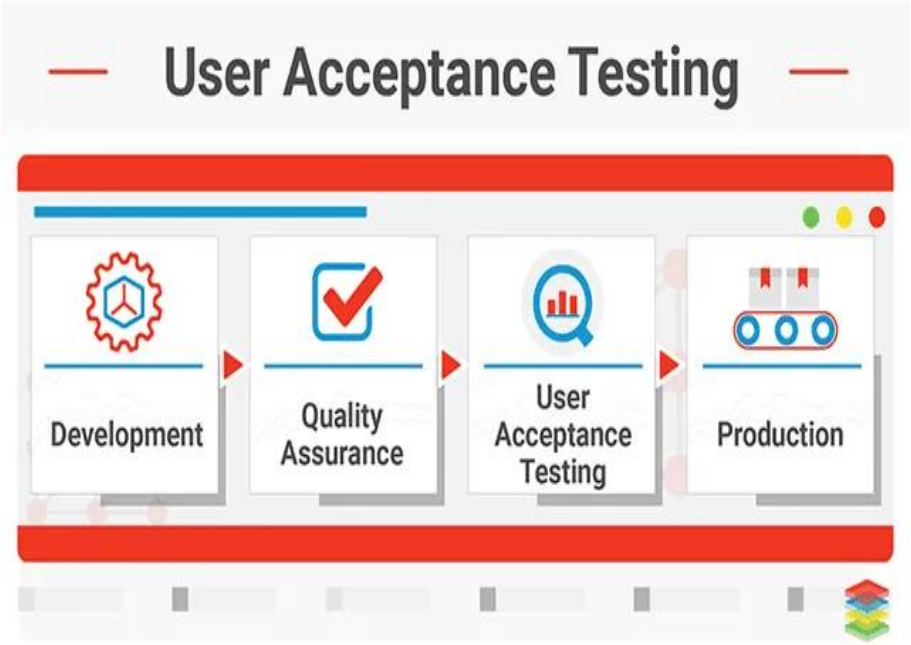


Figure 20. Acceptance test (Xenonstack, 2018).

For instance, in an educational management system, acceptance testing focuses on core functionalities. This means that for teachers, the system must support the ability to organize lists of students and maintain student grades. These actions should trigger system operations that save and display new information across all relevant views instantaneously.

For students, acceptance testing verifies that as soon as a teacher posts grades or attendance, the student can immediately view them on their personal dashboard. This real-time synchronization provides transparency and instant feedback, which is crucial in an interactive learning environment (Myers et al., 2011, p. 284).

4.3 Project Management and Agile Methodologies

This is a specimen of the category User Experience (UX). Many factors come into play, ranging from design to usability to functionality. The key goal of UX is to ensure the product is user-friendly, intuitive, and responsive to the user. Good UX design makes it easy for users to follow, understand what is happening, and use the system through the most efficient, effective, and satisfying steps.

UI refers to the visual design—i.e., the color scheme, layout, typography, and images. UI is the visual side of communication between the product/system and the users. Good UI design means creating an interface that is visually attractive and consistent with how the product is intended to be used, so users can interact easily and efficiently with it.

The interface was created to be simple and easy to use, even for beginners. A minimalist design with a neutral color scheme and straightforward layout was chosen to provide all users with a clear and uncomplicated experience. The transparency of functions and content ensures simplicity and allows for efficient use of the system.

The system was user-centered, with user requirements and expectations taken into account at every stage of the design process. Like a quality dashboard, this interface is intended to allow users to quickly find the features they need, receive immediate feedback on their actions, and use the system without confusion. The overall design promotes ease of use, clarity, and positive user feedback, with performance perceived as good, thereby improving both usability and user experience.

4.4 Technology Framework

4.4.1 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. This runtime introduces competition for the browser and allows developers to build web applications that are fast, scalable, and efficient. By leveraging Node.js for server-side computing, the system can be served between the browser and the server, boosting performance, resulting in a unified language for the web, and contributing an enriched pool of packages and tools for software development (Shah & Soomro, 2017).

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World!');
}).listen(8080);
```

Figure 21. Example of Nodejs (W3S, 2024).

One of the major benefits of Node.js is its aptitude for handling asynchronous operations in a way that does not consume many threads, making it easy to manage millions of concurrent connections with high performance (Harter, 2012). It is a good fit for architectural styles like microservices, which require applications to be composable and scalable (Shah & Soomro, 2017). Being event-driven and non-blocking, it provides a smooth experience, which has made it a preferred backend framework of the 21st century (Pahl, 2015).

I used node.js for my project because it is suitable for real-time web with the event-driven and non blocking I/O architecture, handle many requests without trouble. Furthermore, it also applies the JavaScript language for both client and server, better the consistency source code.

Based on the mechanism single-threaded, asynchronous, it can serve many people at the same time, increasing the response of the system. With the abundant ecology system (npm), it can handle database MySQL, create API with Express.js faster, hold sessions, and file upload.

4.4.2 HTML

HTML (HyperText Markup Language) is the most common language used to create web pages. It is a mechanism for describing parts of a web page—such as text, images, links, or even tables—in this markup language. HTML serves as the foundation of web-based development by arranging and displaying information in an organized and structured manner (W3C, 2021).

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Figure 22. Examle of HTML (W3S, 2021).

One of the best — and hardest-to-miss — aspects of using HTML is that it is structured code, which allows you to easily organize your content so it can be easily managed and understood. It also improves accessibility, making the site more easily comprehensible by search engines and, conversely, assistive technologies (Mozilla Developer Network, 2021). HTML is also cross-browser compatible, meaning web content is consistently displayed across various platforms and devices (Duckett, 2011). Its smooth integration with other technologies like CSS and JavaScript further enhances the user experience by enabling dynamic and visually impressive web applications. "Given the flexibility and last mile optimization [...] of HTML," says Scott Jehl of the

Filament Group, "it's pretty likely that good ol' HTML will continue to be a keystone of modern web development" (W3C, June 2021).

I used HTML for this article because it is the foundation of every website, ensures high compatibility on all devices, and builds clear layouts. It combines well with CSS and JavaScript to make the process smoother, especially when developing long-term applications. HTML will make system maintenance easier.

4.4.3 CSS(Cascading Style Sheets)

CSS is a styling language that describes the design of an HTML document, adding visual effects to the structure and content of web pages. With CSS, authors can style web pages in many ways—for example, creating multi-column layouts, changing colors, adding sounds, and making other improvements to make a web page eye-catching and user-friendly (W3C, 2021).

CSS Example

```css
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Figure 23. Example of CSS (W3S, n.d).

Another advantage of CSS is that it separates content from design, resulting in clean and maintainable code. This division also increases readability and

improves the efficiency of managing and updating styles across multiple pages (W3S, 2024). Moreover, CSS enhances responsiveness, making websites compatible with various devices and screen sizes (MDN Web Docs, 2024). Another benefit is improved performance: reducing inline styles and enhancing page load times results in a much smoother experience, which is why CSS is an important part of the browsing experience (Lark Suite, 2023). CSS has become an integral part of modern web development by providing cascading style properties and making websites visually dynamic and appealing, while ensuring they look good on all platforms (UX Media, 2023).

4.4.4 JavaScript

JavaScript is a language commonly used to develop and interact with websites (MDN Web Docs, n.d.). It allows developers to add features to the page (such as handling events and manipulating the HTML) and interact with users without needing to refresh the page. It is therefore a primary tool for increasing user interest and promoting dynamism in the browsing experience.

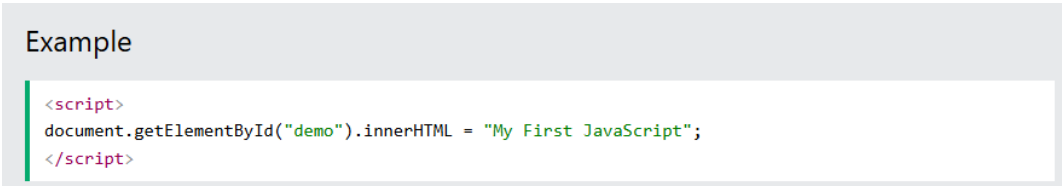


Figure 24. Example of JavaScript (W3S, n.d).

JavaScript handles all data inputs and outputs to avoid refreshing the page in this project. This means you can interact with the server by sending and receiving data through APIs, perform HTTP requests and processing, and manage tasks such as email handling. This enables real-time content updates, making the application more responsive. JavaScript also acts as client-side validation to check user input before sending data to the server, which helps prevent errors and enhances the security of your application. Data handling, user interactions, and real-time updates are made much easier and smoother by JavaScript.

JavaScript makes the application more flexible by not having to reload the page for interactions on the website. The user has a better experience by getting immediate feedback. JavaScript will update the data over time.

4.4.5 Express.js

Express.js is a full-featured web framework for Node.js, designed for quickly building web applications and APIs. It offers a robust toolkit for routing, middleware, and HTTP handling that simplifies the application's lifecycle and makes it easy to scale as the application grows. With its minimalistic and highly flexible approach, Express.js helps clean up complex code and create backend web services using a fundamental standard.

Express.js provides a powerful set of features supporting both client-facing and server-facing JavaScript, aiming to make web application and HTTP API development easy and fast. Its strong capabilities in routing, middleware, and serving HTTP requests enable leaner and more efficient systems. Express's minimalistic and flexible architecture simplifies backend development by making web services feel like Java objects.

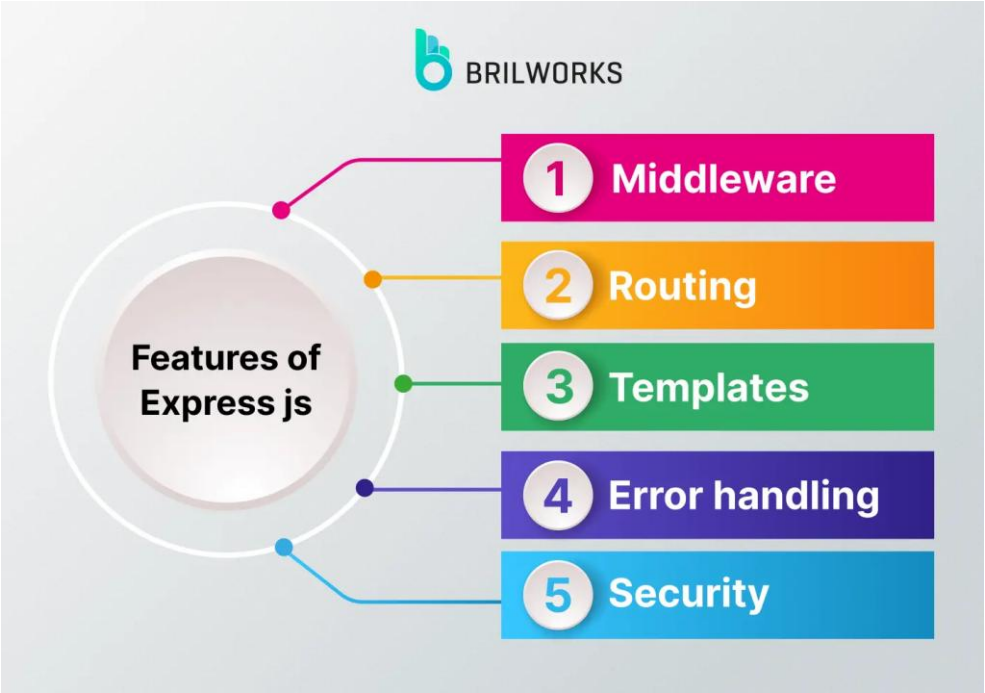


Figure 25. Feature of Express.js (Katariya, 2023).

One of the major benefits of working with Express.js is its capability to start an application in a single JavaScript file, greatly easing setup and configuration. It gives middleware a chance to process incoming HTTP requests, such as parsing POST data, before handling communication between the client and the server-side programs. It also allows developers to define routes for handling different types of HTTP requests in a lean and efficient manner.

Express.js enables easy integration with SQL databases like MySQL, allowing effective data management when query manipulation is needed. Its simplicity, efficiency, and robust middleware-based architecture are key reasons why Express.js attracted me. It truly makes it easy to build fast and scalable network applications.

4.4.6 MySQL

MySQL is an increasingly popular open-source relational database management system (RDBMS) that helps you quickly and easily retrieve data. It is extensively used in web applications where users can store and retrieve data

in a well-organized form to perform various operations like inserting, updating, deleting, and selecting records. MySQL is a popular choice for database-driven application development due to its proven performance, reliability, and ease of use.

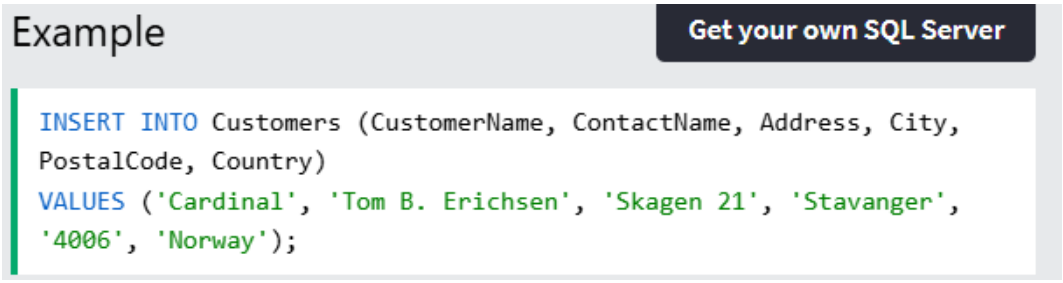


Figure 26. Example of SQL (W3S, n.d).

Express's MySQL package in JavaScript applications is used to initially configure and establish connections with databases to perform operations on them with ease. It allows developers to execute SQL statements, fetch results, and handle the returned data to provide correct responses to the browser. MySQL also offers built-in error-handling techniques to manage errors during query execution. If any request fails or the connection is interrupted, the software can generate an error message and send it back to the client for easy debugging. Because of its organized methodology and strong data-handling capabilities, MySQL is a leading choice for developing back-end or storage-based programs, enabling effective data storage and management for dynamic websites.

4.4.7 Body-Parser

Body-parser is a middleware for Node.js that parses the body of incoming HTTP requests. It offers several advantages, such as making it simple to work with data sent in the payload of requests like POST, PUT, and PATCH. It transforms the request body into a usable JavaScript object, enabling developers to handle incoming data in their applications smoothly.

One of its most important features is JSON parsing. Body-parser processes incoming HTTP requests and normalizes the content so the user can work

with the data in a format that is easy to use and compatible regardless of the content type.

When the server receives an HTTP request, it automatically handles the request and ensures that the JSON from the incoming package is transformed — as mentioned — from JSON to a JavaScript object.

When an Express server receives a request with a content type of application/json, the body-parser passes the data through the server with the request. Before the request is handled by other middleware or routes set up, body-parser ensures all incoming data is in the correct form for the server.

This allows easy access to the data via req.body, enabling developers to use the precise information received. By making the extraction and manipulation of data straightforward, the overall function of the web app is improved, ensuring a good user experience in the interaction between server and client.

5 RESULTS OF DEVELOPMENT

This is all the pages that will be displayed on the website based on the previously written code. It combines backend, frontend and database to produce something like this.

5.1 Registatrion

First, the teacher will go to the registration section to create their own account. Each account will save the teacher's registered information, including email and phone number.

Figure 27. Registration page.

5.2 Login

Once an account is created, teachers will log in to their accounts to see courses, students, and school notifications.

Figure 28. Login page.

5.3 Dashboard

After logging in, we will go to the dashboard section. In this section, we can see announcements from the school, and each announcement will show the time it was posted. We can also access the courses here, and there will be a section to communicate with students.

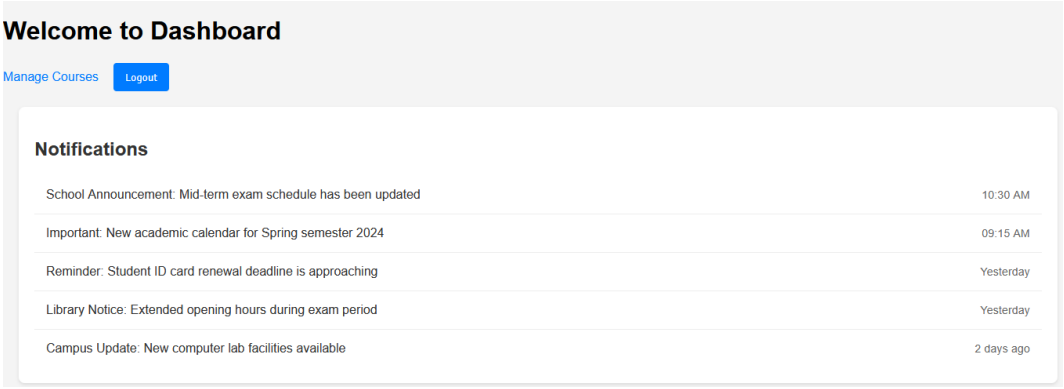


Figure 29. Dashboard page.

5.4 Notifications

In the Notifications section, teachers will receive messages from the school immediately, allowing them to stay updated with time-sensitive alerts based on the priorities set by each district. This enables them to pass on information that students need to know promptly, without any gaps. Having a central system for notifications ensures that teachers do not miss important updates or forget to send crucial information. Moreover, if a notification has not been read, the system ensures that the information is not suppressed, preventing any disruption or negative impact on the learning process. This

simplified system helps keep communication between the school, teachers, and students running efficiently, especially with course management.

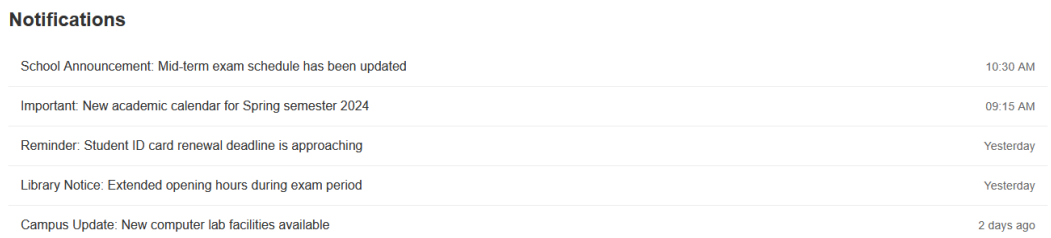


Figure 30. Notification board.

5.5 Student Management

Teachers will see students once they have registered for a course. Through the course, teachers will be able to view students' contact information, see how many students have registered, and easily add or remove students from the course. This will make it easier for users to manage students.

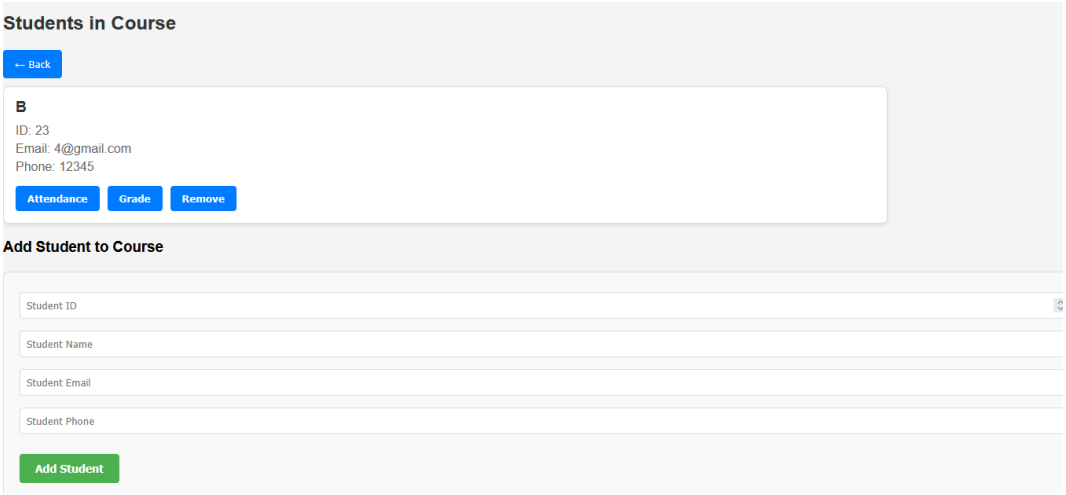


Figure 31. Student management page.

Teachers can also manage their classes smartly by adding or removing courses. A course can be easily added to the system by typing the course name and course code, then hitting the "Add" button. This feature is designed to be easy to use, so teachers shouldn't have any trouble or delays

when using it. Furthermore, it helps teachers efficiently manage course progression, allowing any necessary changes or modifications to be made as needed. Dropping a class is just as simple, enabling teachers to adjust their schedules at any time and keep their teaching routine well-organized.

5.6 Attendance Tracking

It allows teachers to take attendance, view reports, and move students to other classes directly on their handheld devices. Any attendance records will be stored automatically in the database, so nothing will be lost, and everything is always available when needed. Furthermore, this functionality allows teachers to monitor anomalies, such as students who habitually skip class or those who experience problems registering for a class. A well-organized attendance system enables teachers to notice absenteeism patterns, intervene promptly if a student is having trouble, and ensure all students have a chance to get the most out of their education. This holistic model streamlines student management and makes the learning environment more effective as a result.

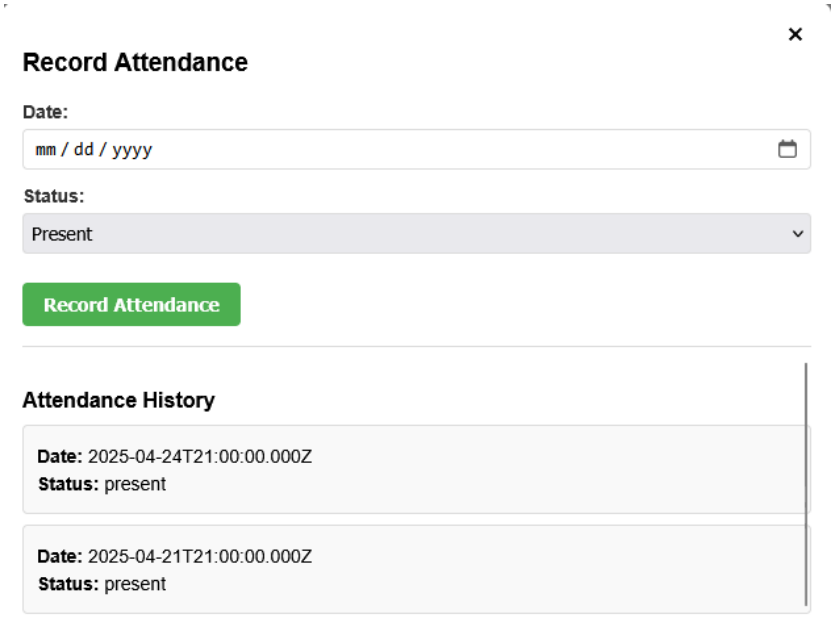


Figure 32. Attendance page.

5.7 Grade submission

Teachers can assign grades to students using the system, making the evaluation process more efficient and transparent. Once a teacher has entered a student's marks, the information is stored in the database, ensuring secure and quick storage of all records. Meanwhile, learners can instantly receive their scores without the need for handouts or physical meetings.

If there is a misunderstanding or error in the grading process, teachers can easily make grade adjustments on their own, without extensive paperwork or an administrative review process. This efficient method is not only convenient but also enhances accuracy for both teachers and students. Furthermore, grades can be saved and retrieved at any time, ensuring that important school records remain accessible whenever needed.

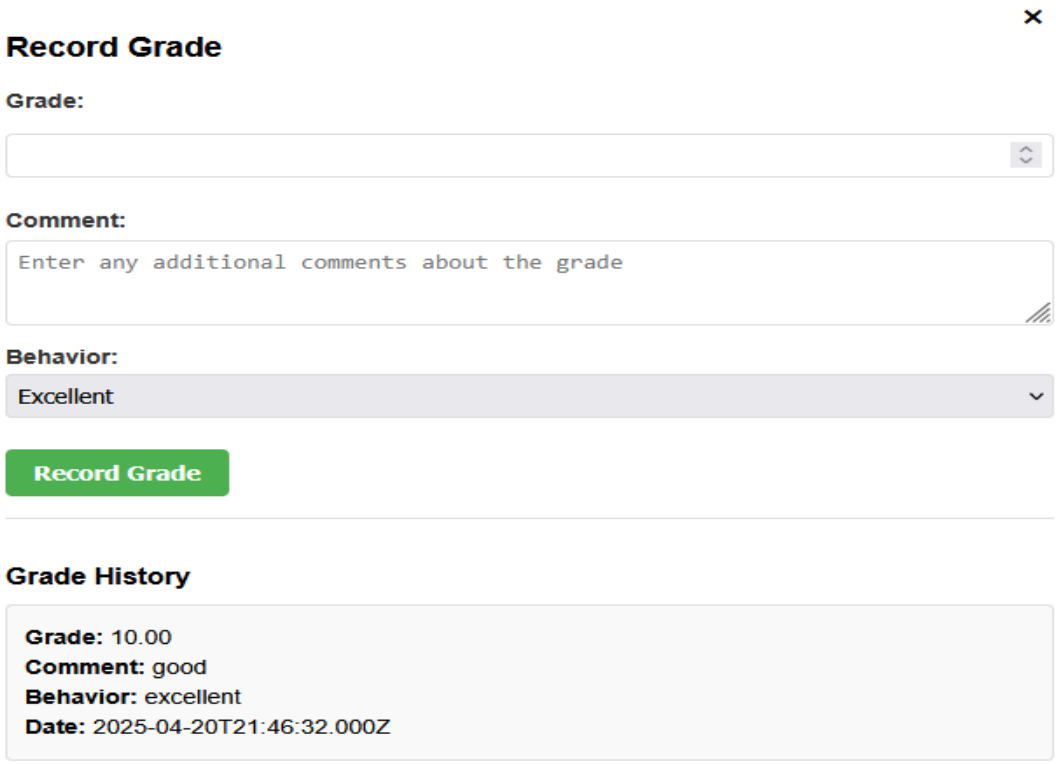


Figure 33. Grade submission page.

5.8 Feedback and Evaluation

During evaluation, teachers will observe students closely as they progress through their learning cycle, monitoring both their academic achievements and behavioral conduct. Constant observation allows teachers to identify issues or provide feedback in real time during the course. The idea is for teachers to input their assessments whenever there is something to address or intervene in, so that each entry can be immediately seen by students. This eliminates the need to wait for scheduled meetings to understand their performance and enables them to make timely adjustments to overcome any shortcomings.

Furthermore, all interactions and assessments will be systematically stored in the database, ensuring the information remains up-to-date and available for future reference. This method not only enhances the efficiency of communication between students and teachers but also provides proper documentation of both academic and behavioral progression.

.

6 EVALUATION AND REFLECTION

6.1 Research Result

Having now completed the research, it is possible to conclude that these findings have been successfully aligned with the theoretical objectives of the study set out at the beginning. The result is a comprehensive management system specifically designed to address the needs of an educational ecosystem—supporting not only administrators but also teachers and learners throughout their educational journey.

This system bridges the gap between instructors and students by eliminating the reliance on third-party platforms for communication, thereby making interactions easier, faster, and centralized within a single application.

One of the main features of the system is its ability to allow teachers to effectively manage student progress, monitor course advancement, and receive timely notifications regarding messages and other communications from the school. Students, in turn, can closely monitor their academic performance, receive teacher feedback, check grades, and communicate directly with instructors through the platform—without the need for face-to-face consultations or external communication tools.

Furthermore, the system makes significant progress in data management. With the assistance of this software solution, both teachers and students can efficiently handle all their data, without relying on outdated methods such as paper-based records or time-consuming manual processes. This minimizes the risk of document misplacement or loss, thereby improving the accuracy, security, and confidentiality of information.

The system also enables teachers to conduct assessments, assign grades, and maintain organized records of student progress. Teachers can easily refer back to or update this information as needed. At the same time, once evaluations or scores are entered into the system, students can instantly

access their results, feedback, and comments—without waiting for physical documentation. This functionality speeds up academic reporting, streamlines communication with instructors, reduces administrative delays, and fosters more open and participatory learning.

In general, this system not only meets the demands of modern school life but also serves as a key component in the modernization of academic processes and control.

Table 1. Testing result of unit test.

Test Case ID	Function	Input	Output	Result
01	Login	Correct account and password	Go to Dashboard	Passed
02	Login	Incorrect account or password	Invalid input	Passed
03	Registration	1@gmail.com	Registration successfully	Passed
04	Registration	1@gmail.com(duplicate)	Email already exist	Passed
05	Add course	Math	Course added successfully	Passed
06	Add course	Math(Duplicate)	Failed to add course	Passed
07	Delete course	Math	Course delete successfully	Passed
08	Delete course	Math(duplicate)	Failed to delete course	Passed

09	Add student	Email, name, studentId,	Student added successfully	Passed
10	Add student	Email, name, studentId(duplicate)	Error adding student	Passed
11	Delete student	Email, name, studentId	Student removed successfully	Passed

Table 2. Testing result of integration test.

Test case ID	Modules	Test Description	Output	Result
01	Login Dashboard	Verify that, after login, the user is redirected to the dashboard page	Once users log in successfully, they are taken to the dashboard	Passed
02	Add course – Data Storage	Check if the course is saved in the system	The new course is shown in the list	Passed
03	Delete course – Data Storage	Check if the course is removed from the system.	The course was removed from the course list.	Passed
04	Add student – Data storage	Check if the student is saved in the system	The student is saved in the system	Passed

05	Delete student – Data Storage	Check if the student is deleted from the system	The student is deleted from the system	Passed
06	Attendance – Data Storage	Check if attendance is saved to the system	Attendance is saved in the system	Passed
07	Score – Data Storage	Check if the score is saved to the system	The score is saved to the system	Passed
08	Comment – Data Storage	Check if the comment is saved in the system	The comment is saved in the system	Passed
09	Behavior – Data Storage	Check if the behavior is saved in the system	Behavior is saved in the system	Passed

Table 3. Testing result of acceptance test.

Test case ID	User Story	Test Description	Output	Result
01	As a teacher, I want to check attendance	Teachers log in, go to course management, select a course,	Attendance data is saved and reviewed accurately.	Passed

		and check each student's attendance		
02	As a teacher, I want to enter student scores, comments, behavior	Teacher logs in, goes to course management, selects a course, views the student list, and marks scores, comments, and behavior	Score, comment, and behavior data are saved and reviewed accurately	Passed
03	As a teacher, I want to see the notification from school	Notifications from the admin (school) are retrieved from data storage	Reviewed notifications are accurate.	Passed

6.2 Research Methods

A qualitative method was used in undertaking this study to explore in depth the use of IT in the administration of schools. Instead of conducting extensive surveys or inquiries to arrive at an answer, the research was primarily

based on the collection and analysis of existing academic, research, and practical information—particularly through direct development and practice-enabling efforts (Pacheco et al, 2025).

By independently examining literature and analyzing technology frameworks, a digital system was designed and implemented to address specific requirements in learning and educational management. Experiential in nature, this process led to the development of a website intended to improve the student experience and assist educational institutions and educators in providing better administration of academic processes.

The qualitative approach was an appropriate method for the purpose of the study due to its flexibility and independence. It facilitated a deeper engagement with the content, allowing personal interaction with both the technical and theoretical elements of the study (RJosefsson, P & Jää-Aro, K & Lundmark, S & Mutvei, A., 2019).

The findings included a thorough overview of how technology can enhance communication, data storage, and learning management in schools, supported by an analysis of various sources. These findings directly influenced the creation of a user-friendly and responsive platform that simplifies communication and administrative tasks between teachers and students without the need for additional third-party products (NCBI, 2023).

The qualitative research methodology led to a more profound understanding of the field and to the development of a system that is functional, practical, and meets the needs of its users (MECS Press, 2018).

6.3 Difficulties In The Research Process

One of the first challenges I faced in my research was the problem of obsolete and irrelevant data. Most of the references I found were outdated for

today's educational scene. Moreover, the sheer volume of available sources made it difficult to determine which were the most accurate, reliable, and relevant to the task. I spent a lot of time sifting through different readings and sources, trying to find information that corresponded with the needs of my research and remained current today. It was a slow process that required a lot of thinking and wisdom.

Second, there was the challenge of managing the volume of information—many times I struggled to piece together and organize everything into a coherent process. Sometimes I got confused and went off in various directions that were not part of the initial research. This highlighted the necessity of staying focused and frequently returning to the aims and objectives of the study to stay on track.

Time was also an issue. I was not just analyzing massive amounts of data; I was simultaneously building and designing a working website that materialized the insights and solutions I developed during my PhD. Striking a balance between the technicalities of web development and the huge demands of scholarly research was very difficult, especially while juggling other jobs and projects. It all came down to planning, prioritizing, and persistence to ensure everything was done properly and on time.

The web development process was not without its issues either. I had to overcome several coding mistakes and random bugs, which took a significant amount of time to debug. Some of the tech and programming languages I used were ones I had not worked with in a long time, so I can only say: horrible experience! I had to spend more time in the docs and tutorials to refresh my knowledge and get back up to speed. That said, going through this process gave me valuable experience, helping me think things through and dive deeper than ever into both the technical and literature sides of my project.

REFERENCES

Anderson, B. (2023) – *Integration Testing: Definitions, Types, and Best Practices* From https://briananderson2209.medium.com/integration-testing-definitions-types-and-best-practices-f2e6047ee448

Bates, T. (2015). *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning.* Retrived 30.3.2025. From 15.05.2025 from https://tecfa.unige.ch/guides/e-books/Bates-Teaching-in-a-Digital-Age-compressed.pdf

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide. Addison-Wesley*. From https://www.researchgate.net/publication/234785986_Unified_Modeling_Language_User_Guide_The_2nd_Edition_Addison-Wesley_Object_Technology_Series

Burnstein, I. (2003). *Practical Software Testing: A Process-Oriented Approach*. *Springer*. [Page 115]. From: https://svkarthikwinner.wordpress.com/wp-content/uploads/2015/07/practical-software-testing.pdf

Crispin, L. & Gregory, J. (2009) – *Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley*. Page 143–144. From https://scrummalaysia.com/media/attachments/2020/03/18/agile_testing_-_a_practical_guide_for_testers_and_agile_teams.pdf

Coursebox AI. (n.d.). *What is a Student Management System?* Retrieved from https://www.coursebox.ai/blog/what-is-a-student-management-system

Codebox (2022) – *Unit Test: What is, Adventages and How works?* From https://codenboxautomationlab.com/unit-testing-what-is-advantages-how-with-example/

CSS tutorial, w3school. From https://www.w3schools.com/css/default.asp

Davis, F. D. (1989). *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly, 13(3), 319-340.* Retrieved from https://www.researchgate.net/publication/200085965_Perceived_Usefulness_Perceived_Ease_of_Use_and_User_Acceptance_of_Information_Technology

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Retrieved from https://sites.math.duke.edu/courses/math_everywhere/assets/techRefs/HTML%20and%20CSS-%20Design%20and%20Build%20Websites_Jon%20Duckett_2011.pdf

Garrison, D. R., & Anderson, T. (2003). *E-Learning in the 21st Century: A Community of Inquiry Framework for Research and Practice. Routledge*. Retrieved from https://www.researchgate.net/publication/287556984_E-Learning_in_the_21st_century_A_framework_for_research_and_practice_Second_edition

HelioCampus. (2017). *Effective Data Management in Higher Education: Obstacles and Solutions*. Retrieved from https://cdn2.hubspot.net/hubfs/2492978/HelioCampus_2017/Docs/HelioCampus%20Effective%20Data%20Management%20eBook%20Final.pdf

Harter, P. (2012). *Node.js in Action. Manning Publications.* Retrieved from https://www.manning.com/books/node-js-in-action

HTML. (2021). Retrieved from https://www.w3.org/TR/html52/

Jorgensen, P. C. (2013). *Software Testing: A Craftsman's Approach (4th ed.). CRC Press.* [Page 58]. From: https://malenezi.github.io/malenezi/SE401/Books/Software-Testing-A-Craftsman-s-Approach-Fourth-Edition-Paul-C-Jorgensen.pdf

Josefsson, P., Jää-Aro, K., Lundmark, S., Berrez, A. (2019). *THE IMPLEMENTATION OF DIGITAL TOOLS IN TEACHING: AQUALITATIVE CASE STUDY AT A SWEDISH PRIMARY SCHOOL.2382-2387.* From https://www.researchgate.net/publication/334676803_THE_IMPLEMENTATION_OF_DIGITAL_TOOLS_IN_TEACHING_A_QUALITATIVE_CASE_STUDY_AT_A_SWEDISH_PRIMARY_SCHOOL

Katariya, L. (2023) – *What is Express.js? A comprehensive Guide to Beginners.* From https://www.brilworks.com/blog/what-is-express-js-comprehensive-guide-to-beginners/

Long, P., Siemens, G. (2011). *Penetrating the Fog: Analytics in Learning and Education.* EDUCAUSE Review, 46(5),30-32. From: https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education?_gl=1*6s8txs*_gcl_au*NTYxNjA2Nzg3LjE3NDgzNzY5NzguMTk0NzczNjA4LjE3NDgzNzcwMzIuMTc0ODM3NzAzMQ.

Lark Suite. (2023). *Cascading Style Sheet (CSS).* Retrieved from https://www.larksuite.com/en_us/topics/marketing-glossary/cascading-style-sheet-css

Myers, Glenford J., Sandler, Corey, & Badgett, Tom. (2011). *The Art of Software Testing* (*3rd ed.).* Wiley. Page 281, 284. https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf

MDN Web Docs. (n.d.). *MVC (Model-View-Controller)*. Retrieved from https://developer.mozilla.org/en-US/docs/Glossary/MVC

MDN Web Docs. (n.d.). *Server-side web frameworks – Learn web development,* from https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks

MDN Web Docs. (2024). *Responsive Design - CSS*. *Mozilla*. Retrieved April 30, 2025, from https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design

Mozilla Developer Network. (2021*). HTML: Hypertext Markup Language*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML

Mozilla, (n.d). *What is JavaScript* (From https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Nicol, D., & Macfarlane-Dick, D. (2006). *Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice.* https://www.reap.ac.uk/reap/public/papers/DN_SHE_Final.pdf

NCBI (2023) – *Digital transformation in higher education: A qualitative evaluative study of a hybrid virtual format using a smart classroom system*.Heliyon,Volume 9, Issue 6. From https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10248111/

Nguyen, T. & Tran, H. & Nguyen, M. (2023). *Empowering Education: Exploring the Potential of Artificial Intelligence; Chapter 9 -Artificial Intelligence (AI) in Teaching and Learning: A Comprehensive Review.* From: https://www.researchgate.net/publication/374508985_Empowering_Education_Exploring_the_Potential_of_Artificial_Intelligence_Chapter_9_-Artificial_Intelligence_AI_in_Teaching_and_Learning_A_Comprehensive_Review

Omonije, A. (2024). *Agile Methodology: A Comprehensive Impact on Modern Business Operations. International Journal of Science andResearch (IJSR)*. From https://www.researchgate.net/publication/377979833_Agile_Methodology_A_Comprehensive_Impact_on_Modern_Business_Operations

Oranga, J., Matere, A. (2023). *Qualitative Research: Essence, Types and Advantages. OALib. 10. 1-9.* From https://www.researchgate.net/publication/376739890_Qualitative_Research_Essence_Types_and_Advantages

Pappas, C. (2015)**.** *The Top LMS Statistics and Facts For 2015 You Need To Know. eLearning Industry*. https://elearningindustry.com/top-lms-statistics-and-facts-for-2015

Pacheco, A.,Yupanqui R., Mogrovejo, D., Garay, J., Uribe-Hernandez, Y. (2025). *Impact of digitization on educational management: Results of the introduction of a learning management system in a traditional school context*. Computers in Human Behavior Reports, Volume 17, March 2025, Article 100592. From https://doi.org/10.1016/j.chbr.2025.100592

Prieto, M., Zapata, A., Dominguez, V. (2011). *Data Mining Learning Objects. In Handbook of Educational Data Mining* (pp. 3-19). CRC Press. From https://www.researchgate.net/publication/262047702_Data_Mining_Learning_Objects_In_Handbook_of_Educational_Data_Mining

Pahl, C. (2015). *Cloud Computing: Concepts, Technology &Architecture. Springer*. Retrieved from https://ptgmedia.pearsoncmg.com/images/9780133387520/samplepages/0133387526.pdf

Shah, H., & Soomro, T. R. (2017). *Node.js challenges in implementation.* ResearchGate. https://www.researchgate.net/publication/318310544_Nodejs_Challenges_in_Implementation

Sheldon, R (12 September, 2023) *What is model-view-controller (MVC)?* Retrieved from https://www.techtarget.com/whatis/definition/model-view-controller-MVC

Slawek-Polczynska, A. (2020), *Agile software development process.* From https://soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/

UX Media. (2023). *CSS: The Backbone of Modern Web Design.* Retrieved from https://uxmedia.io/blog/what-is-the-role-of-css-in-web-design/

User Acceptance Testing Tools, Types and Best Practices. From https://medium.com/digital-transformation-and-platform-engineering/user-acceptance-testing-tools-types-and-best-practices-5090e8e5b2f5

Wikipedia contributors. (n.d.). *Model–view–controller*. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Model–view–controller

W3S. (n.d, -a) *HTML tutorial*. Retrieved 28.04.2025. From https://www.w3schools.com/html/default.asp

W3C. (2021). *CSS: Cascading Style Sheets*. Retrieved 28.04.2025. From https://www.w3.org/TR/css-2021/

W3S. (2024). *CSS Introduction*. From https://www.w3schools.com/css/css_intro.asp

W3S (n.d-b) *JavaScript Where to*. From https://www.w3schools.com/js/js_whereto.asp

W3S (n.d) *SQL tutorial*. From https://www.w3schools.com/sql/sql_insert.asp

W3S (n.d) *Nodejs tutorial*. From https://www.w3schools.com/nodejs/default.asp

APPENDICES

APPENDIX 1. What is an appendix

You can include, for example, the questionnaire you used in the survey and other material relevant to the survey.

Appendices may also include material that the contractor wishes to keep confidential, in which case the appendix will be omitted from the thesis.

APPENDIX 2. Titling and numbering of appendices

If there is only one appendix, change the title of the APPENDICES to APPENDIX: NAME OF APPENDIX (Style name Appendix X). If there are two or more appendices, the appendices are numbered and titled. Where reference is made to an appendix in the text, this is done in accordance with the referencing conventions.