

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



SECURE WEB REPORT

Group: 3

Topic: Student Management Website
Instructor: Mr. Bui Trong Tung
Mr. Do Ba Lam
Students: Le Duc Dung 20214952
Nguyen Quoc Huy 20210427
Nguyen Hoang Anh 20214946
Duong Hong Nam 20214967

HANOI - 06/2024

TABLE OF CONTENTS

1	Introduction	2
1.1	Project Overview	2
1.2	Purpose and Goals	2
1.3	Scope	2
2	Website Features	3
2.1	Online Class Management	3
2.1.1	Admin	3
2.1.2	Teacher	5
2.1.3	Student	6
2.2	Attendance Checking	7
2.2.1	Teacher	7
2.2.2	Student	8
2.3	Multiple Choice Test	9
2.3.1	Teacher	9
2.3.2	Student	11
3	Software Structure and Database Design	12
3.1	Software Structure	12
3.2	Database Schema	13
3.2.1	Class Management Tables	13
3.2.2	User Authentication Table: tbltoken	14
3.2.3	Examination and Evaluation Tables	14
3.2.4	Attendance Management Tables	15
4	Information Security Techniques	17
4.1	Authentication and Password Management	17
4.2	Session Management and Access Control	19
4.3	Input Handling and Output Cleaning	19
4.4	Event Logging	20
5	Deployment	21
6	Security Scan Results	22
6.1	Security Analysis	22
6.2	Maintainability Assessment	24
6.3	Reliability Evaluation	25
6.4	Duplications	26
7	Conclusion	27

1 Introduction

1.1 Project Overview

This project is a comprehensive Student Management System designed to facilitate teachers in managing student attendance and administering tests. The system integrates modern technologies to streamline educational management processes, making them more efficient and user-friendly. A notable feature is the QR code-based attendance system, which simplifies and accelerates the process of marking attendance.

1.2 Purpose and Goals

The main objective of this project is to create a secure and efficient platform for student management that can be easily used by teachers. The specific goals include:

- Enabling teachers to take student attendance quickly and accurately using QR codes.
- Providing a platform for teachers to create, manage, and grade tests for students.
- Ensuring the security and privacy of student data through robust security measures.

1.3 Scope

The project encompasses the following features and security measures:

- Features Implemented:
 - QR code-based attendance system.
 - Test creation and management functionality.
 - Student performance tracking.
 - User-friendly interface for teachers and students.
- Security Measures Applied:
 - Secure authentication and authorization mechanisms.
 - Data encryption for sensitive information.
 - Input validation to prevent common security threats like SQL injection and XSS.
 - Preventing brute force attacks using reCAPTCHA v2
 - Properly assign permissions for different users such as students, teachers, and admins.

2 Website Features

This section provides a detailed description of the features available on our educational platform, categorized into online class management, attendance checking, and multiple-choice tests. Each category includes functionalities specific to Admins, Teachers, and Students.

2.1 Online Class Management

This feature allows users to manage classes and related information effectively.

2.1.1 Admin

- Create and manage class schedules.

This feature allows admins to have the right to create classes and manage the list of classes they have created, such as assigning teachers to which classes, viewing class total students, adding or deleting students to classes, and handling related errors to class,...

The screenshot shows the 'Add Class' form in the Student Management System. The form is titled 'Add Class' and has a sidebar on the left with navigation links: SMS, Admin, ADMIN, Dashboard, Class, Add Class, Manage Classes, Students, Teachers, Notice, Public Notice, and Pages. The form fields are: Class Name (Badminton I), Room (NTD), and Teacher (Duong Nam). There is an 'Add' button at the bottom of the form. The footer of the page says 'Student Management System'.

Figure 1: Add Class

The screenshot shows the 'Manage Class' table in the Student Management System. The table is titled 'Manage Class' and has a sidebar on the left with navigation links: SMS, Admin, ADMIN, Dashboard, Class, Add Class, Manage Classes, Students, Teachers, Notice, Public Notice, and Pages. The table has columns: No., Name, Code, Teacher, Creation Time, and Action. The table contains 4 rows of data. The footer of the page says 'Student Management System'.

No.	Name	Code	Teacher	Creation Time	Action
1	Web Programming	8d2n1d	Duong Nam	2024-01-15 17:42:47	Edit Delete
2	Cryptography	abc81h	Nguyen Be	2024-01-12 17:42:41	Edit Delete
3	Introduction to Programming	9dn2dc	Nguyen Anh	2024-01-10 17:45:35	Edit Delete
4	Soft Skills	72nd1	Nguyen Anh	2024-01-10 17:42:14	Edit Delete

Figure 2: Manage Class

- Add and manage teacher.

This feature allows admins to have the right to create accounts for teachers and manage information of the accounts they have created, edit detailed information of a teacher account that needs to be updated or There are errors in information or deletion of that teacher’s account when the teacher is no longer working in the system.

Figure 3: Add Teacher

No.	Teacher ID	Name	Username	Email	Date	Action
1	20002	Duong Nam	nam	c@gmail.com	2024-05-18 08:05:42	Edit Delete View
2	20001	Nguyen Be	be	b@gmail.com	2024-05-16 09:35:52	Edit Delete View
3	20000	Nguyen Anh	anh	hwsignuprandomthing@gmail.com	2024-05-15 10:18:58	Edit Delete View

Figure 4: Manage Teacher

- Add and manage student.

This feature allows admins to have the right to create accounts for students and manage information of the accounts they have created, edit detailed information of a student account that needs to be updated or There are errors in the information, or the student’s account is deleted when the student is no longer in the system.

SMS

> Add Student

> Manage Students

Teachers

Notice

Public Notice

Pages

Email

Gender

Choose Gender

Student ID

Contact Number

User Name

Password

Add

Figure 5: Add Student

SMS

Admin

ADMIN

Dashboard

Class

Students

> Add Student

> Manage Students

Teachers

Notice

Public Notice

Pages

Manage Students

View all Students

No.	Student ID	Name	Username	Email	Date	Action
1	20210427	Nguyen Quoc Huy	nguyenhuy	nguyenquochuy712@gmail.com	2024-05-15 11:14:35	Edit Delete
2	20230001	Nguyen C	nguyenc	nguyenc@gmail.com	2022-01-19 22:24:52	Edit Delete
3	20220001	Tran B	tranb	tranb@gmail.com	2022-01-16 13:23:33	Edit Delete
4	20210001	Nguyen A	nguyena	nguyena@gmail.com	2022-01-13 21:09:04	Edit Delete

First > Prev > Next > Last

Student Management System

Figure 6: Manage Student

2.1.2 Teacher

- View class schedules.

This function allows teachers to have an overview of the classes they have been assigned by the system, and view class information such as the number of students in the class, class code,...

SMS

Nguyen Anh

TEACHER

Dashboard

View Notice

My Class

My Test

My Class

View all Classes

No.	Name	Code	Student	Test	Attendance	Action
1	Introduction to Programming	9sh26c	4	3	0	Edit
2	Soft Skills	72nc1	4	1	0	Edit

First > Prev > Next > Last

Student Management System

Figure 7: View Class

- Manage class materials and announcements.

Teachers can also edit the class code, and add class announcements so students

can grasp information such as which classes are absent, which classes are pushed forward or changed classrooms, test time, etc. ...

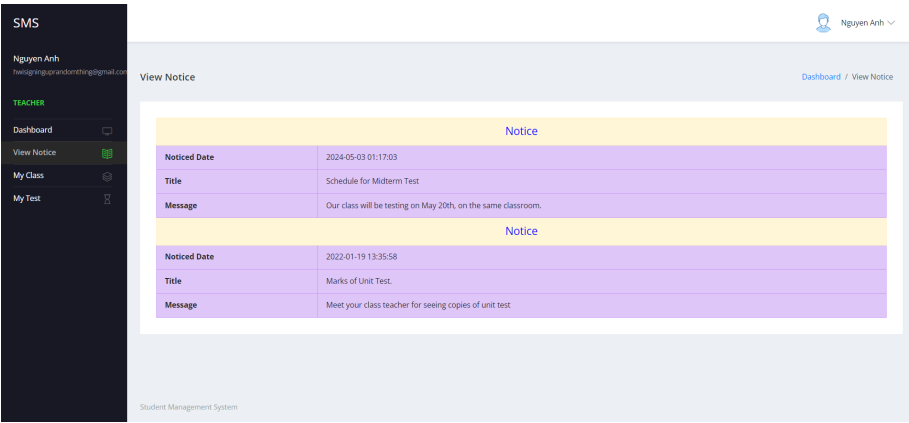


Figure 8: View Notice

2.1.3 Student

- View class schedules and details, join classes.

Students have an overview of the list of classes they have participated in, and can also participate in other classes through the class code and approval of the teacher in charge of that class.

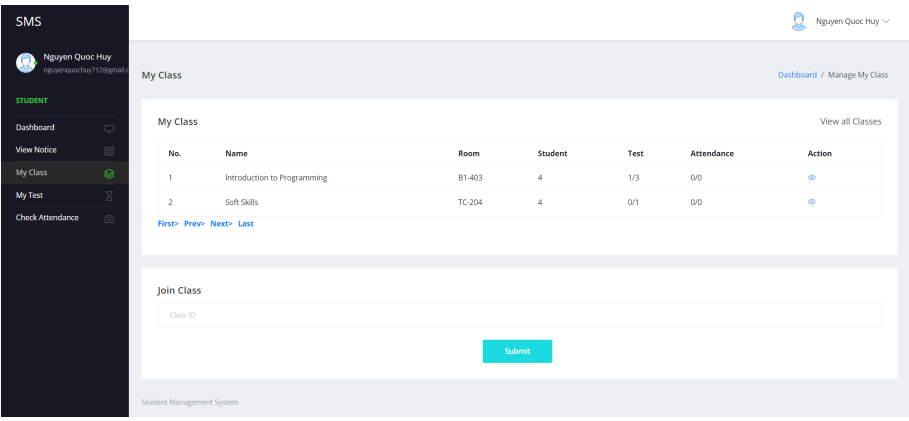


Figure 9: View and join class

- Receive notifications and announcements.

Students can also view announcements of the class they have attended to grasp necessary information about the class such as whether this class is off, whether they need to change classrooms, or whether there are upcoming tests.

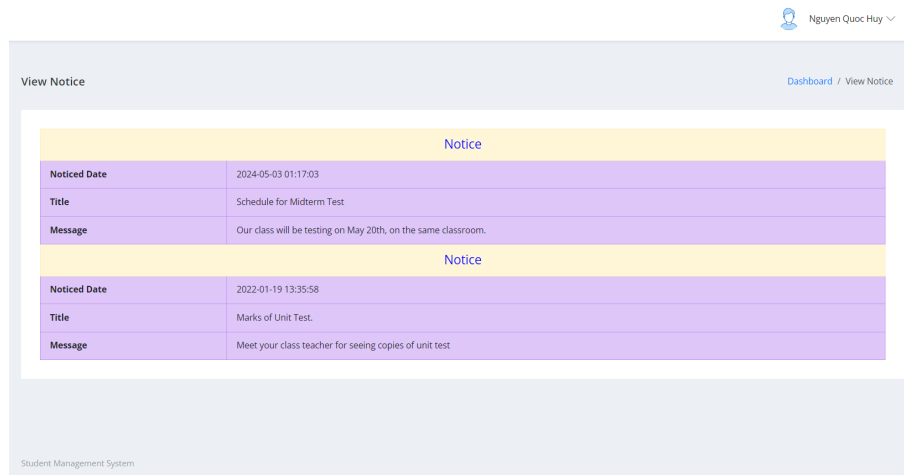


Figure 10: View Notice

2.2 Attendance Checking

This feature ensures that attendance is tracked and managed efficiently.

2.2.1 Teacher

- Generate QR code for attendance checking.

This function allows teachers to create QR codes to serve the attendance of students in that class. Each QR code will correspond to 1 record and in a lesson the teacher can arbitrarily create 1 or many attendance records to monitor students' attendance process more closely.

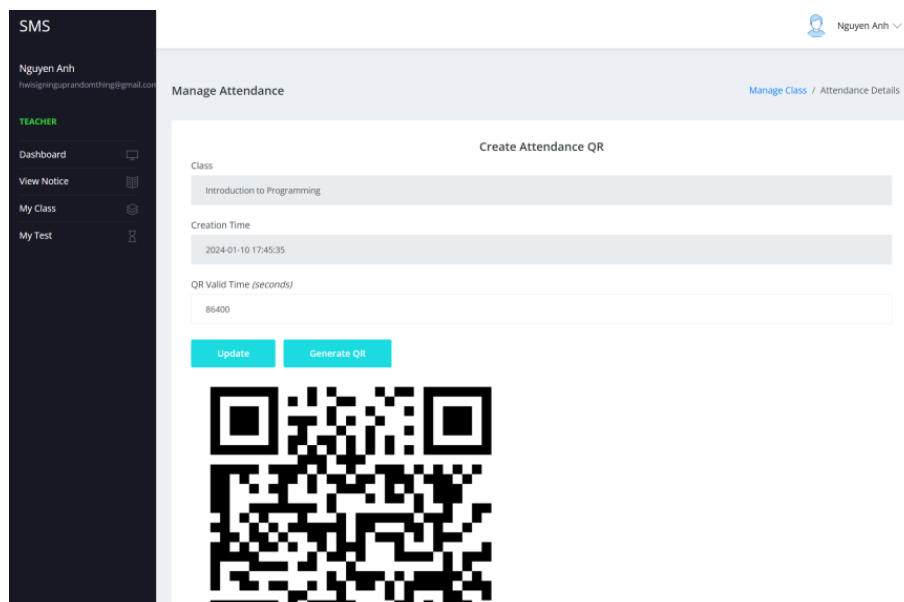


Figure 11: Generate QR code

- Review attendance records for their classes.

After students use the QR code to take attendance, the attendance list corresponding

to each record the teacher has created in the class will appear. Teachers can view and manage this list to manage class attendance

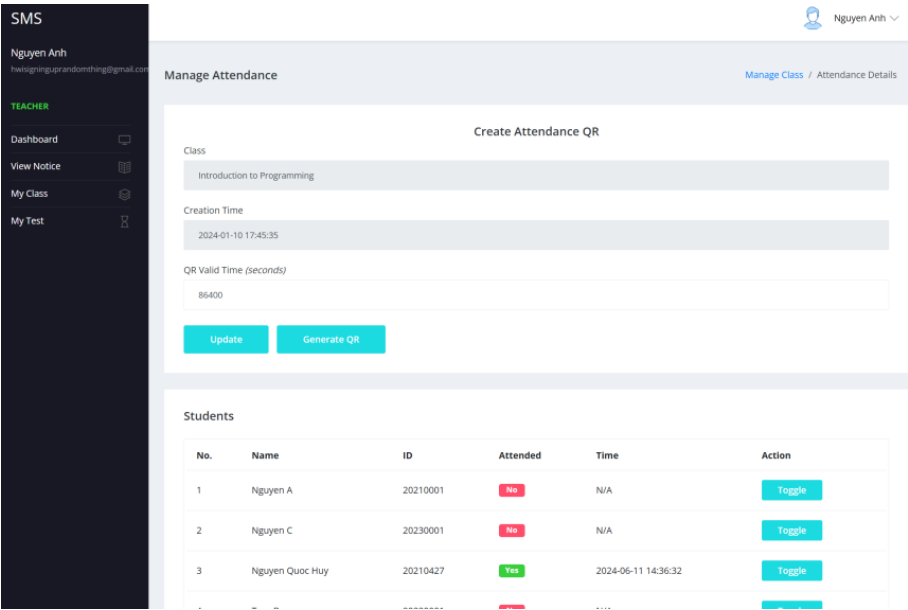


Figure 12: View Attendance List

- Follow up with students on attendance issues.

At the same time, teachers can also add, edit, and delete attendance records manually if cheating or problems arise during the student's attendance process.

2.2.2 Student

- Check personal attendance records.

Students can use the camera and the QR code scanning function that appears inside the class interface they have joined to scan the QR code created by the teacher to take attendance for the class they are studying.

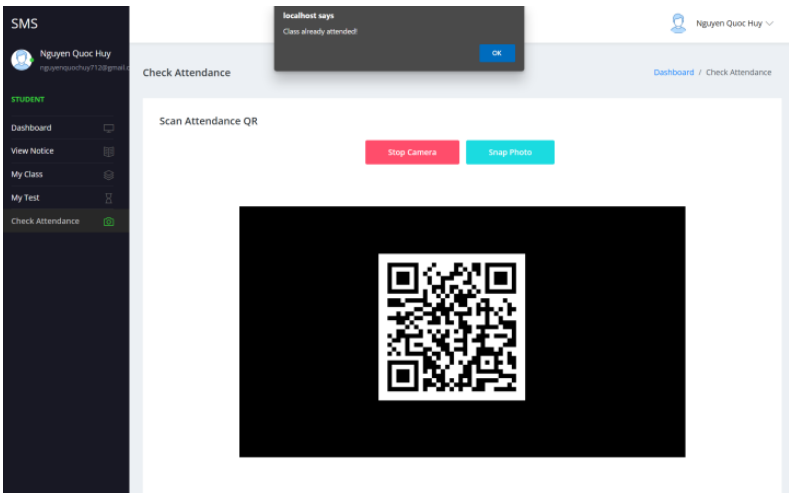


Figure 13: Checking Attendance

- Keep track of attendance status.

After taking attendance, students can also see their attendance status in that class, with each record whether they have taken attendance or not or whether the status is still recorded as absent.

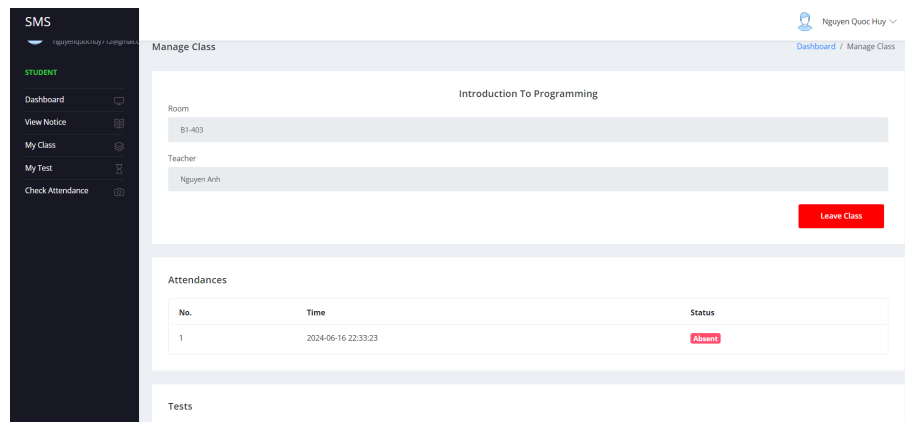


Figure 14: View Attendance

- Report any issues with recorded attendance.

If any technical problems occur during the attendance process, students can also complain directly to the teacher or through the website system.

2.3 Multiple Choice Test

This feature allows for creating, taking, and managing multiple-choice tests. The system supports reuse question for multiple tests, and 2 question type (Single answer or Multiple answers)

2.3.1 Teacher

- Create tests and quizzes for their subjects.

Teachers can create tests, manage test time and test start time, add and edit questions in the test for the class they are managing, or cancel the test when there are technical problems happen.

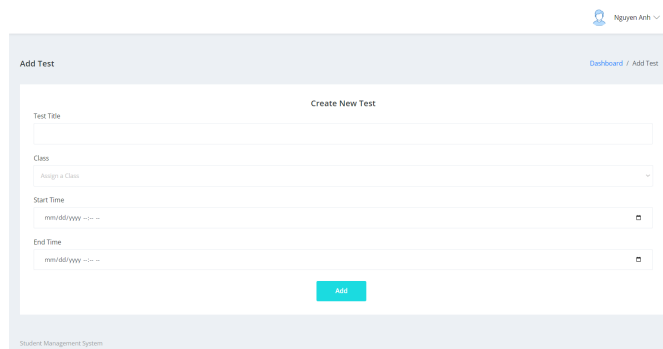


Figure 15: Create Test

- Add or edit questions in the question bank.

Teachers can also add or delete questions in the question bank, which is used to set questions for tests of classes that teachers undertake.

SMS
 Nguyen Anh
 hieu@nguyendarandthing@gmail.com

TEACHER

- Dashboard
- View Notice
- My Class
- My Test

Edit Question Manage Questions / Edit Test Question

Question #80002

Question

Which are the correct answers?

Answer	Content	Correct Answer(s)
A	Not this	<input type="checkbox"/>
B	This	<input checked="" type="checkbox"/>
C	This	<input checked="" type="checkbox"/>
D	Not this	<input type="checkbox"/>

Allow Multiple choices

Yes

[Save Change](#) [Delete](#)

Figure 16: Add Question to Question Bank

Questions Bank Dashboard / Manage Created Questions

View Questions

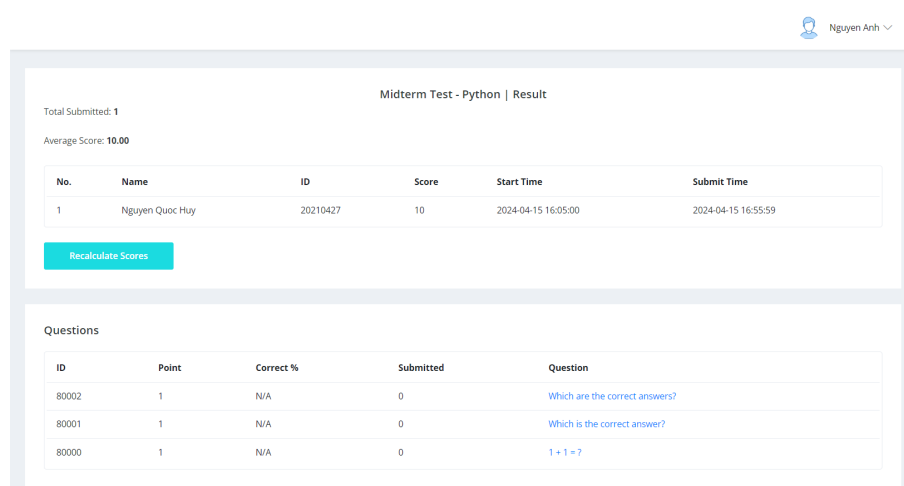
ID	Type	Content	Action
80002	Single	Which are the correct answers?	⌕
80001	Single	Which is the correct answer?	⌕
80000	Single	1 + 1 = ?	⌕

[First>](#) [Prev>](#) [Next>](#) [Last](#)

Figure 17: Manage Question Bank

- Analyze student test results for performance assessment.

After students complete the test, teachers can also have an overview and statistics on the results of the just completed test and edit the results manually if any technical problems happen during student work.



No.	Name	ID	Score	Start Time	Submit Time
1	Nguyen Quoc Huy	20210427	10	2024-04-15 16:05:00	2024-04-15 16:55:59

[Recalculate Scores](#)

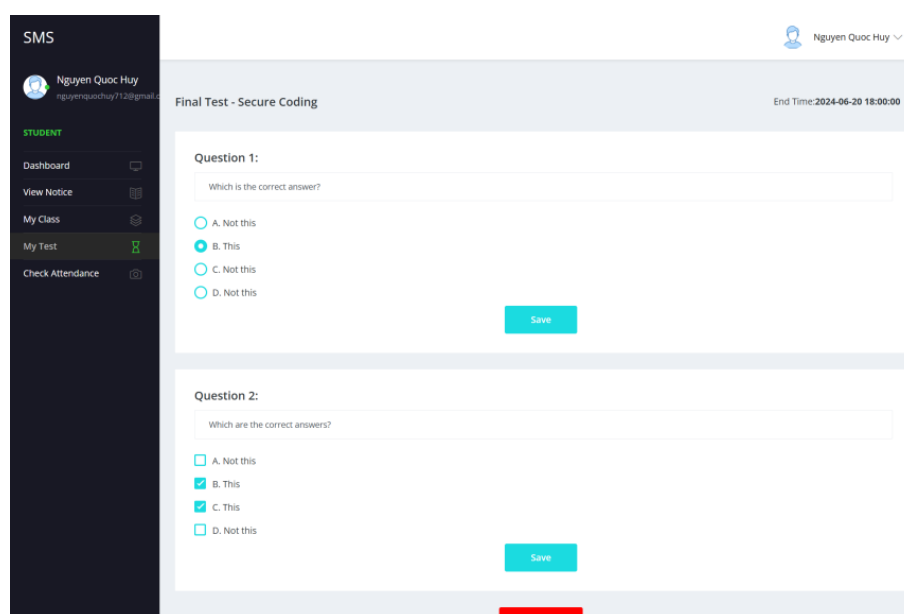
ID	Point	Correct %	Submitted	Question
80002	1	N/A	0	Which are the correct answers?
80001	1	N/A	0	Which is the correct answer?
80000	1	N/A	0	1 + 1 = ?

Figure 18: View Test Result

2.3.2 Student

- Take multiple choice tests online.

Students can participate in tests that have been created by teachers in the class they have joined after receiving notification about the start time of the previous test.



SMS

Nguyen Quoc Huy
nguyenquochuy712@gmail.com

STUDENT

- Dashboard
- View Notice
- My Class
- My Test
- Check Attendance

Final Test - Secure Coding

End Time: 2024-06-20 18:00:00

Question 1:

Which is the correct answer?

☐ A. Not this
☒ B. This
☐ C. Not this
☐ D. Not this

[Save](#)

Question 2:

Which are the correct answers?

☐ A. Not this
☒ B. This
☒ C. This
☐ D. Not this

[Save](#)

Figure 19: Taking Test

- Immediately feedback on test performance.

Students can make complaints about any technique problems they encounter during the exam.

- Review correct answers and explanations post-test.

After completing the test, students can also review the overall results of the test and the answers to the questions if the teacher allows.

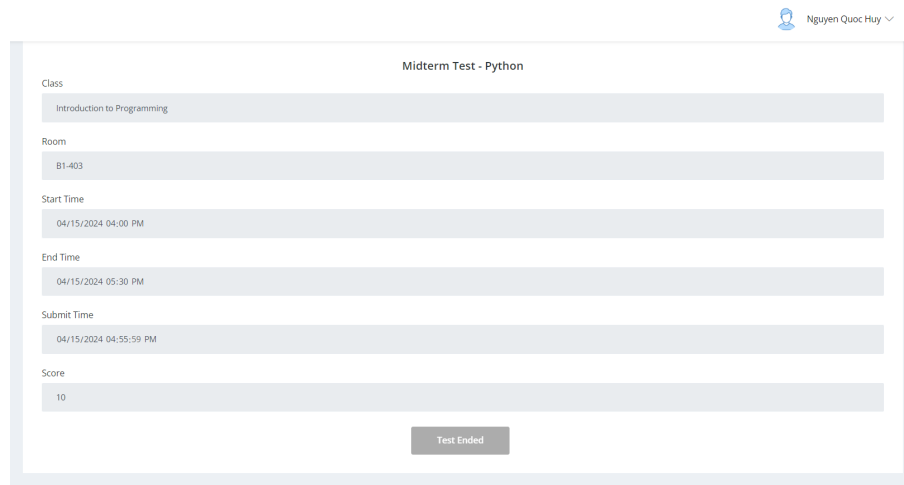


Figure 20: View Test Result

3 Software Structure and Database Design

3.1 Software Structure

The Student Management website utilizes a three-tier architecture for optimal performance, security, and maintainability. Here's a breakdown of each layer:

1. Presentation Layer (Front-end):

- This layer focuses on user interaction and visual representation.
- Technologies like HTML, CSS, and PHP create a user-friendly interface.
- The front-end interacts with the back-end through PHP includes to exchange data and perform actions.

2. Business Logic Layer (Back-end):

- This layer handles the system's core functionalities, including data processing, business logic implementation, information security, and communication with the database.
- PHP is chosen as the server-side scripting language for its robustness and extensive web development capabilities.
- The back-end handles tasks like user authentication, data validation, class management, notice creation and delivery, and student management.
- It interacts with the database to store, retrieve, and manipulate class data, notice information, and other relevant information.

3. Database Layer (Database):

- This layer stores all system data in a structured format for efficient retrieval and manipulation.
- MySQL serves as the chosen relational database management system (RDBMS) due to its popularity, scalability, and ease of use.

- A well-defined database schema is designed to store class information, notice details, student records, and teacher data.
- The back-end utilizes PHP libraries or MySQL APIs to interact with the database and perform CRUD (Create, Read, Update, Delete) operations on the data.

This layered architecture ensures the separation of concerns, promoting code maintainability, security, and scalability. The front-end focuses on presentation, the back-end handles business logic, security and the database securely stores the system's data.

3.2 Database Schema

Our web project's database consists of 16 tables, which are organized as follows:

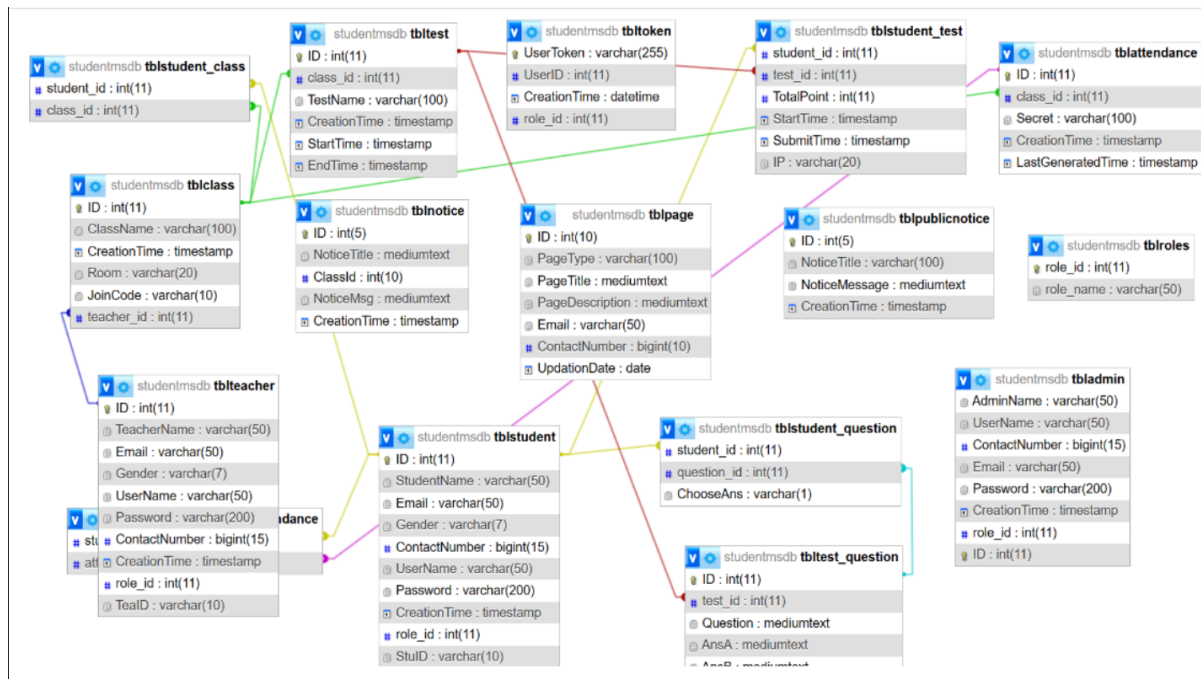


Figure 21: Overview of the database schema

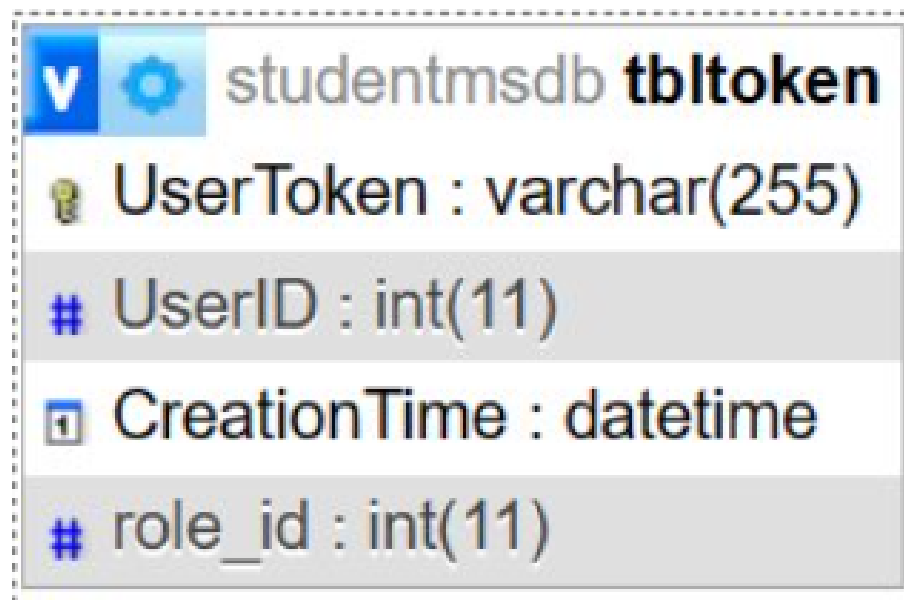
3.2.1 Class Management Tables

- tbladmin: Contains administrative account details.
- tblstudent: Stores student account information.
- tblteacher: Holds teacher account details.
- tblclass: Manages class information.
- tblpage: Used for page management in the website.
- tblnotice: Contains notices from the admin for students and teachers.
- tblpublicnotice: Stores public notices to be displayed system-wide.

3.2.2 User Authentication Table: tbltoken

The tbltoken table is pivotal for managing user authentication within our system. It associates each user ID with a unique token and records the token's creation time. The server validates each client request by verifying the correct user ID, token, and ensuring the token is within a valid time interval.

- **UserToken:** A 128-bit randomly generated token for each user after logging in successfully, stored as a variable character string with a maximum length of 255 characters.
- **UserID:** The user's identification number, stored as an integer.
- **Creation Time:** The date and time when the token was created, stored in datetime format.
- **Role ID:** An integer representing the user's role, which determines access permissions within the system. 1 for admins, 2 for teachers and 3 for students



	studentmsdb	tbltoken
🔑	UserToken	: varchar(255)
#	UserID	: int(11)
📅	CreationTime	: datetime
#	role_id	: int(11)

Figure 22: tbltoken

3.2.3 Examination and Evaluation Tables

The examination and evaluation component of our database is designed to manage and track student assessments effectively. The following tables are central to this process:

- **tblstudent_test:** This table connects students with their assigned tests and records the start and end times of each test attempt.
 - **ID:** A unique identifier for each record.
 - **Student ID:** References the student's ID in the tblstudent table.

- **Test ID:** Corresponds to the test’s ID in the `tbltest` table.
- **StartTime:** Timestamp of when the student started the test.
- **EndTime:** Timestamp of when the student completed the test.
- **Passwd:** The password used for test access, stored as a variable character string.
- `tbltest_question`: Holds the questions for each test, along with possible answers and the points for the correct answer.
 - **ID:** Unique identifier for each question record.
 - **Test ID:** Associates the question with a specific test.
 - **Question:** The content of the question, stored as medium text.
 - **AnsA to AnsD:** The possible answers for the question.
 - **Point:** The number of points awarded for the correct answer.
- `tblstudent_test_question`: Tracks the responses of students to each question in a test.
 - **ID:** A unique identifier for each student’s response record.
 - **Student ID:** Links to the student’s ID in the `tblstudent` table.
 - **Test ID:** Connects to the test’s ID in the `tbltest` table.
 - **Question ID:** Refers to the question’s ID in the `tbltest_question` table.
 - **Choose:** The option chosen by the student, stored as a tiny integer.
- `tblclass`: Manages information related to classes.
 - **ID:** Unique identifier for each class.
 - **ClassName:** The name of the class.
 - **CreationTime:** Timestamp of when the class was created.
 - **RoomCode:** The code for the classroom.
 - **JoinYear:** The year when the class was formed.

3.2.4 Attendance Management Tables

The attendance management system is structured to efficiently record and monitor student presence in classes. The following tables form the backbone of this system:

- `tblstudent`: Contains comprehensive details about students.
 - **ID:** The unique identifier for each student.
 - **StudentName:** The full name of the student.
 - **Email:** The email address of the student.
 - **Gender:** The gender of the student.

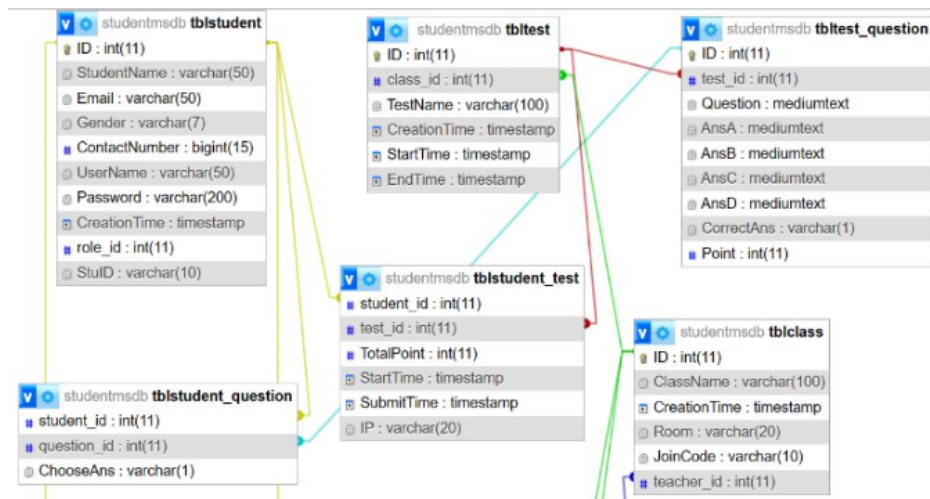


Figure 23: Examination and evaluation tables schema

- **ContactNumber**: The contact number of the student.
- **Address**: The residential address of the student.
- **Password**: The password for the student’s account.
- **CreationTime**: The timestamp of the account creation.
- **Role ID**: The role identifier, defining student access levels.
- **Std ID**: A secondary identifier for the student.
- **tblclass**: Manages class-related information.
 - **ID**: Unique identifier for each class.
 - **ClassName**: The name of the class.
 - **CreationTime**: Timestamp of when the class was created.
 - **Room**: The room number or location of the class.
 - **JoinCode**: The code used by students to join the class.
 - **Teacher ID**: The identifier for the class teacher.
- **tblstudent_attendance**: Links student IDs with their attendance records.
 - **Student ID**: The student’s unique identifier.
 - **Attendance ID**: The unique identifier for the attendance record.
- **tblattendance**: Records the attendance details for each class session.
 - **ID**: Unique identifier for each attendance record.
 - **Class ID**: Links to the class’s unique identifier.
 - **Student ID**: Associates the attendance record with a student.
 - **CreationTime**: The timestamp of when the attendance was recorded.
 - **Secret**: A secret code associated with the attendance record.
 - **LastGeneratedTime**: The last time the attendance record was generated.

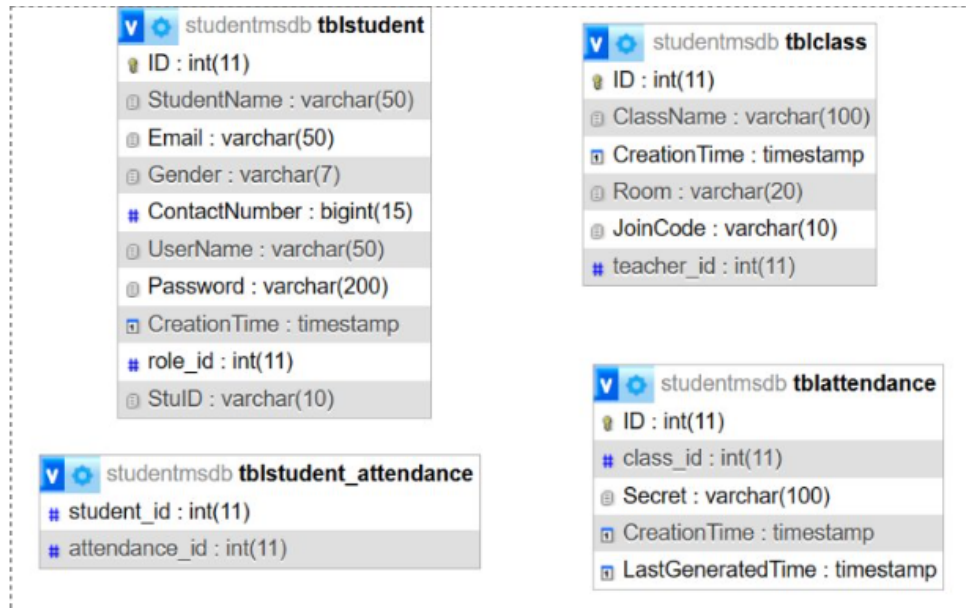


Figure 24: Attendance management tables schema

4 Information Security Techniques

4.1 Authentication and Password Management

All teacher and student accounts are created by the admin at the beginning with a secure password that adheres to robust complexity requirements. Each password must contain at least 8 characters, including 1 special character, 1 capital letter, and 1 number to enhance security.

The user ID and hashed password, along with salts, are securely stored in the database to protect against potential data breaches.

Furthermore, the implementation of re-captcha v2 at the login page serves as a defense mechanism against brute-force attacks by verifying that login attempts are made by humans rather than automated bots. To prevent brute-force attacks, implement reCAPTCHA v2 on the login page (see Figure 25). In addition to these measures, the system employs a two-factor authentication (2FA) mechanism. This adds an extra layer of security by requiring users to provide two different authentication factors to verify themselves. This significantly decreases the risk of unauthorized access. Enable Two-Factor Authentication (2FA) for an additional layer of security (see Figure 26).

Moreover, in case a user forgets their password, the system provides a secure method for resetting it. The client can request a password reset which triggers an OTP (One-Time Password) to be sent to their registered email address. This OTP must be entered along with a new password to complete the reset process, ensuring that only the legitimate account owner can reset the password. Allow clients to reset their password using a One-Time Password (OTP) sent to their registered email (see Figure 27). The important thing is the system allows only ONE attempt to avoid brute-force attack. So if the user entered wrong OTP, they must contact the admin for resetting password.



Hello! Let's get started

Sign in to continue.



I'm not a robot



reCAPTCHA
[Privacy](#) - [Terms](#)

Login

Figure 25: Login Page with reCAPTCHA

Two Factor Authentication

Disabled

Update

Figure 26: Two-Factor Authentication (2FA)

Student Management System

Hi,

Use the following OTP to complete your 2FA Login procedures.

948406

If this is not you, please do not share this OTP.

Figure 27: Password Reset via OTP

4.2 Session Management and Access Control

Effective session management is crucial for maintaining the security and integrity of user interactions with the system. The following measures are implemented to ensure robust session management:

- Each user session is uniquely identified by a combination of the user ID and a secure token which is 128-bit randomly generated. The user ID and this token is stored in the user's cookie, allowing the server to recognize and validate each session.
- The lifespan of each session is capped at 2 hours, after which it expires minimize the risk of unauthorized access over extended periods of inactivity.
- When a user logs out, their session cookie is immediately disabled, effectively terminating the session and further securing against unauthorized reuse.
- The 'HTTPOnly' attribute is set for all session cookies, which helps prevent session hijacking by ensuring that cookies are not accessible via client-side scripts.
- Access rights are managed by verifying the user ID and token attached to each request. The server performs this check with every request to ensure that users can only access pages and resources they are authorized to view.
- Every request is checked against the 'tbltoken' database for a valid user ID, token and a corresponding role ID so that any attempt to access an invalid page, or an unauthorized action results in the user being redirected to the login page, thereby preventing unauthorized access to restricted areas of the system. This ensures that no information is leaked or accessed by unauthorized entities

These measures collectively form a comprehensive session management strategy that safeguards against common security threats and ensures that user sessions are managed securely throughout their lifecycle.

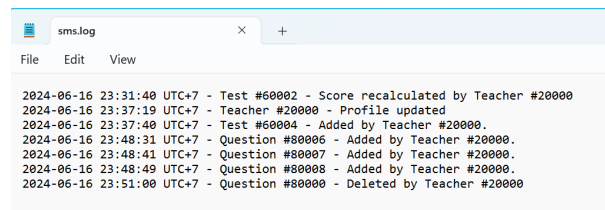
4.3 Input Handling and Output Cleaning

To maintain the security and integrity of our application, we implement stringent input handling and output cleaning measures. These measures are designed to thwart common security threats such as SQL Injection (SQLi) and Cross-Site Scripting (XSS).

- **Prepared Statements:** We utilize prepared statements, a feature provided by PDO in PHP, to prevent SQLi. This method ensures that user input is treated as data rather than executable code in our SQL queries.
- **Input Length Restriction:** We impose limitations on the length of user input across our forms and API endpoints. This not only helps in preventing buffer overflow attacks but also aids in maintaining data integrity and storage efficiency.
- **Output Escaping:** To prevent XSS, we ensure that any user input that will be outputted to the browser is properly escaped. Special HTML characters are converted to their respective HTML entities using functions like `htmlentities` or `htmlspecialchars` in PHP, ensuring they are not interpreted as code by the browser.

4.4 Event Logging

To keeping track of all security and auditing related event, we also save each action event inside a sms.log file.



```
sms.log
File Edit View

2024-06-16 23:31:40 UTC+7 - Test #60002 - Score recalculated by Teacher #20000
2024-06-16 23:37:19 UTC+7 - Teacher #20000 - Profile updated
2024-06-16 23:37:40 UTC+7 - Test #60004 - Added by Teacher #20000.
2024-06-16 23:48:31 UTC+7 - Question #80006 - Added by Teacher #20000.
2024-06-16 23:48:41 UTC+7 - Question #80007 - Added by Teacher #20000.
2024-06-16 23:48:49 UTC+7 - Question #80008 - Added by Teacher #20000.
2024-06-16 23:51:00 UTC+7 - Question #80008 - Deleted by Teacher #20000
```

Figure 28: Log Output

By adhering to these practices, we ensure that our application handles user input securely and outputs data cleanly, thereby safeguarding against potential vulnerabilities.

5 Deployment

How to run this Project:

- Download and extract the project zip file
- Require PHP, XAMPP, Composer to be installed
- Paste inside root directory(for xampp xampp/htdocs, for wamp wamp/www, for lamp var/www/Html)
- Open PHPMyAdmin (<http://localhost/phpmyadmin>)
- Import studentmsdb.sql file (given inside the zip package in SQL file folder)
- Require ';extension=gd' to be enable in 'php.ini'
- Run 'composer install' inside the directory

For the credentials of admin:

- username: admin
- password: Password@123

After access to the admin account, you can create teacher and student accounts on your own in the 'Add teacher' and 'Add student' features of the admin role, then you can login to the account that you created for testing the features.

6 Security Scan Results

The security scan of our project, conducted through SonarCloud has yielded critical insights into the current state of our codebase. The following subsections detail the findings and their implications for our project’s overall code quality and security posture.

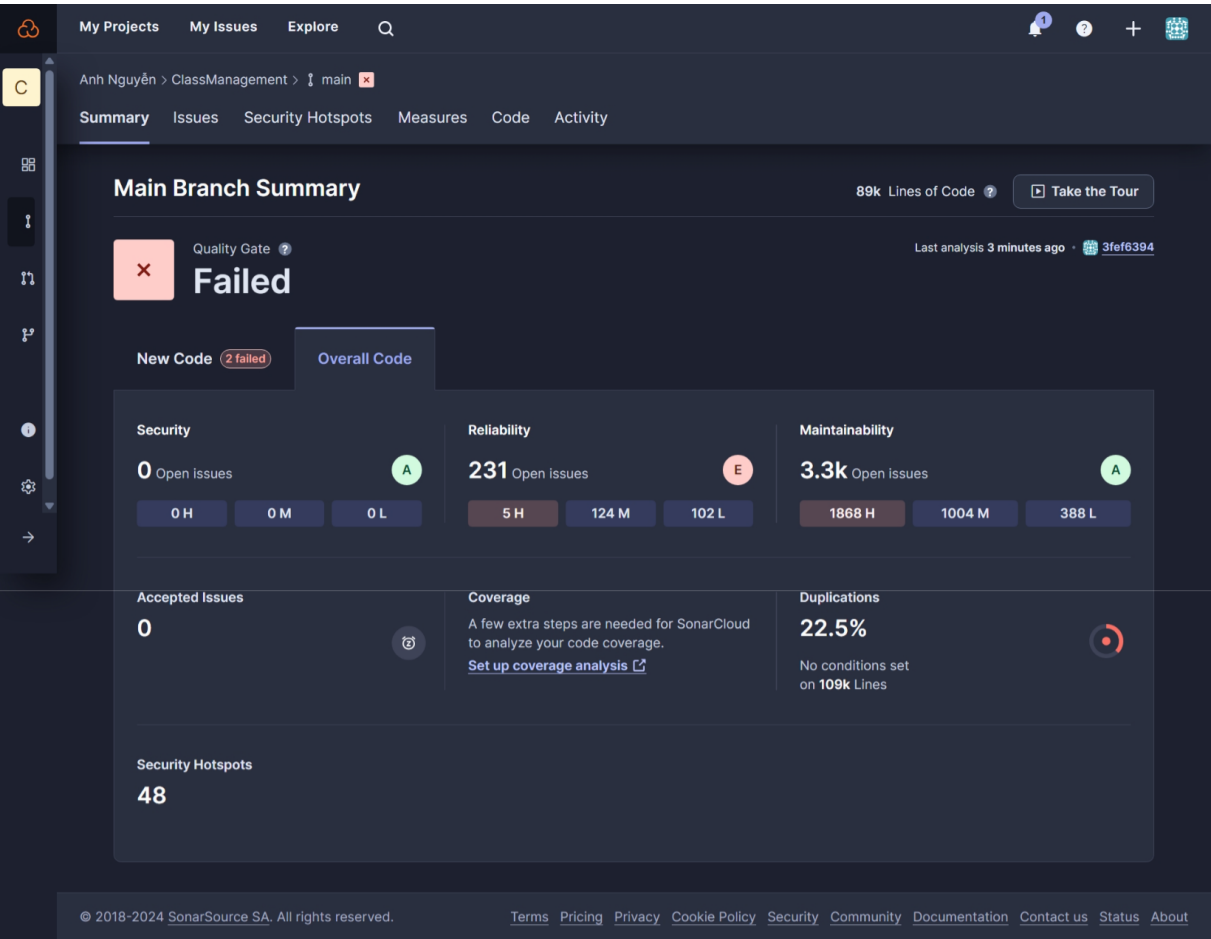


Figure 29: Overview of the scan result in SonarCloud

6.1 Security Analysis

The security analysis of our project, as conducted by SonarCloud, provides a comprehensive assessment of the codebase’s security posture. The following points elaborate on the key metrics and their significance:

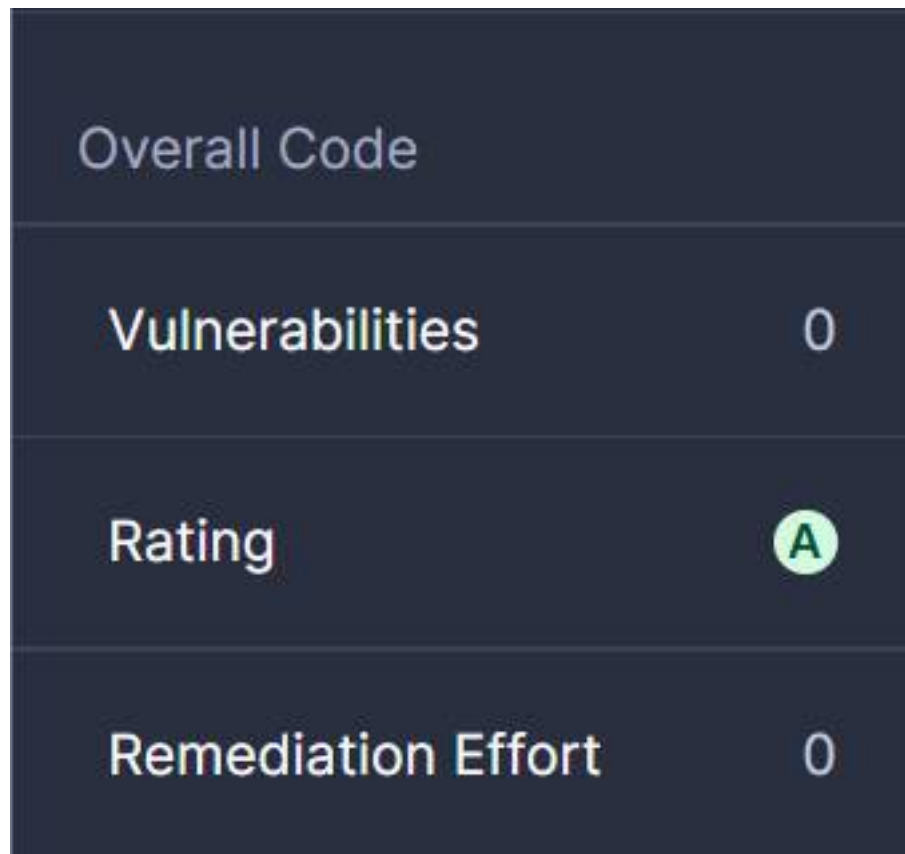


Figure 30: The security scan result in SonarCloud

- **Overall Code Security:** Our project has achieved an exemplary security status with no vulnerabilities detected in the scanned code. This indicates a robust application that adheres to best practices in security coding standards.
- **Vulnerabilities:** A count of zero vulnerabilities signifies that the code is well-protected against known security risks, and there have been no instances of security weaknesses that could be exploited by threat actors.
- **Security Rating:** With a top-tier rating of 'A', the project stands as a testament to our commitment to security. This rating reflects a codebase that is not only secure by design but also maintained with vigilance to prevent the introduction of security flaws.
- **Remediation Effort:** The remediation effort metric stands at zero, underscoring that no additional work is required to address security concerns. This is indicative of a proactive approach to security, where potential issues are resolved promptly, and preventative measures are effectively implemented.

These metrics collectively demonstrate a high level of security maturity within our project. Our development practices, continuous integration processes, and adherence to security guidelines have culminated in a codebase that is secure, reliable, and maintainable.

6.2 Maintainability Assessment

The maintainability assessment of our project, as evaluated by SonarCloud, provides a granular look into the codebase's long-term sustainability and ease of management. The following points detail the specific maintainability concerns and their implications:

Overall Code	
Code Smells	3,267
Debt	37d
Debt Ratio	0.7%
Rating	A
Effort to Reach A	0

Figure 31: The maintainability scan result in SonarCloud

- **Code Smells:** A total of 3,267 code smells were detected, which are indicative of small maintainability issues that, while not currently impacting functionality, could complicate future code modifications or enhancements. The majority of these are related to unnecessary parentheses and spacing, or the practice of not starting 'if' statements on a new line.
- **Technical Debt:** Represented as 37 days (37d), technical debt quantifies the extra

development work that arises from choosing an easy solution now instead of using a better approach that would take longer. Addressing the identified code smells would reduce this debt, streamlining future development efforts.

- **Debt Ratio:** At 0.7 percent, the debt ratio is relatively low, suggesting that the codebase is in good health overall. However, even a small percentage can grow if not addressed, emphasizing the importance of continual code quality improvements.
- **Maintainability Rating:** The project has received an 'A' rating for maintainability, which is the highest possible score. This reflects a codebase that is well-structured and largely free of technical debt that would hinder future maintenance.
- **Effort to Reach 'A':** No additional effort is required to maintain the 'A' rating, indicating that the current state of the codebase meets the highest standards of maintainability. Nonetheless, addressing the minor issues related to code style and structure will further enhance the code's readability and maintainability.

In conclusion, while the maintainability rating is strong, attention to the finer details of code style, such as the placement of parentheses, spacing, and control statement formatting, will ensure that the code remains clean, understandable, and easy to work with for all contributors.

6.3 Reliability Evaluation

The reliability evaluation of our project, as determined by SonarCloud, provides an insight into the stability and error-proneness of our codebase. The following points detail the specific reliability concerns and their implications:

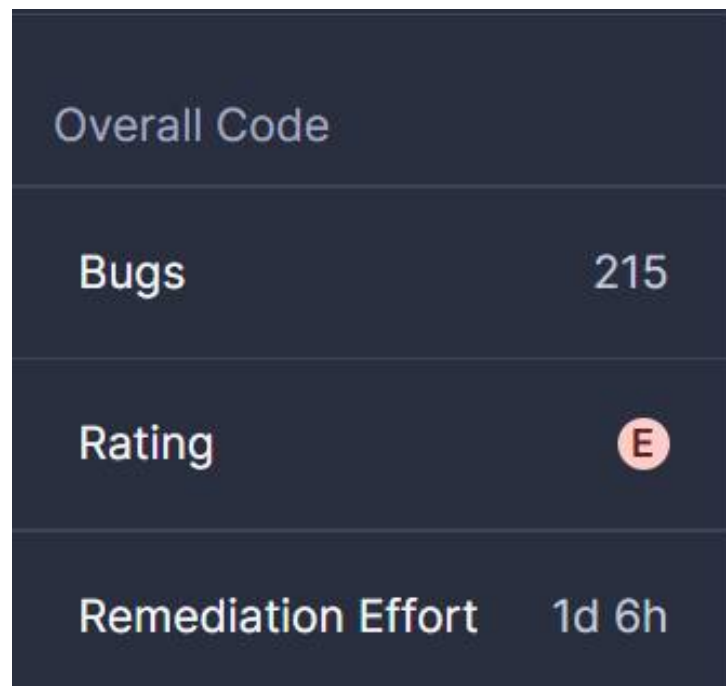


Figure 32: The reliability scan result in SonarCloud

- **Bugs:** The analysis identified a total of 215 bugs. While this number may seem daunting, it primarily encompasses instances where the code uses "include" and "require" statements instead of their "_once" counterparts.
- **Reliability Rating:** The project has been assigned a rating of 'E', which is indicative of a codebase that, while functional, has areas that need refinement to enhance stability and prevent potential runtime errors.
- **Impact of Identified Issues:** The use of "include" and "require" without "_once" can lead to code being included or required multiple times, which may result in redeclaration errors and unexpected behaviors. By replacing these statements with "include_once" and "require_once", we ensure that each file is included only once, thus preventing such issues.
- **Best Practices:** Adopting "include_once" and "require_once" is considered a best practice when the same file might be included or required multiple times during execution. This change not only improves reliability but also aligns the code with industry standards.

In conclusion, the reliability evaluation underscores the need for minor yet impactful modifications to our inclusion statements. Implementing these changes will significantly improve the reliability of our application and align our practices with professional coding standards.

6.4 Duplications

The duplications metric stands at **22.5%**, with **3861 duplicated blocks** detected. This level of code redundancy is suboptimal and calls for a thorough review and refactoring to promote modularity and eliminate unnecessary repetition in the codebase.

In conclusion, while the scan results showcase areas of excellence, particularly in security issue resolution, they also highlight critical aspects requiring attention, such as reliability and code duplications. Addressing these will be pivotal in elevating our project's quality and securing its success.

7 Conclusion

In conclusion, the Student Management System project successfully addresses the essential needs of educational institutions by providing a robust, user-friendly platform for managing student attendance and administering tests. The system leverages modern technologies to streamline these processes, significantly enhancing the efficiency and effectiveness of educational management.

Key Achievements:

- **Efficient Attendance Management:** The integration of a QR code-based attendance system simplifies the process of taking attendance, making it quick and accurate. This innovative approach reduces the administrative burden on teachers and ensures reliable attendance records.
- **Comprehensive Test Management:** The platform enables teachers to create, manage, and grade tests with ease, providing a seamless way to assess student performance. This feature supports various test formats and automates the grading process, saving valuable time for educators.
- **Enhanced User Experience:** The system is designed with a user-friendly interface that ensures an intuitive and engaging experience for both teachers and students. The inclusion of features such as the "Forgot Password" option and personalized welcome messages further enhances usability.
- **Robust Security Measures:** The project implements comprehensive security measures to protect user data and prevent unauthorized access. The use of secure authentication mechanisms, data encryption, input validation, and reCAPTCHA for brute force prevention ensures that the system is both secure and reliable.

Lesson Learned:

- **Security as a Priority:** Implementing robust security measures from the outset is crucial in protecting sensitive user data and maintaining the trust of users.
- **Flexibility in Development:** Being flexible and open to changes based on new findings or user feedback allowed the project to adapt and improve continuously.
- **Role Definition:** Clearly defining roles and responsibilities helped in efficient task management and ensured accountability.