Name: Trịnh Tuấn Phong
Student ID: GCS220282
Subject code/Name: COMP1841
Type: Coursework
Submission Date: 29/11/2025
Assignment Name: Web Programming 1

Topic:

https://drive.google.com/drive/folders/1yXbqi_mjY6xBM0wNs591lDMqhJf0UcbE

Student Forum Website Report

Table of Contents

1. Introduction

In the current internet age, students need online tools to support their learning and group work. Realizing the lack of a dedicated space to exchange lessons, I developed the Student Forum Website project. This report provides an overview of the platform, designed to meet the real needs of students at Greenwich Vietnam.

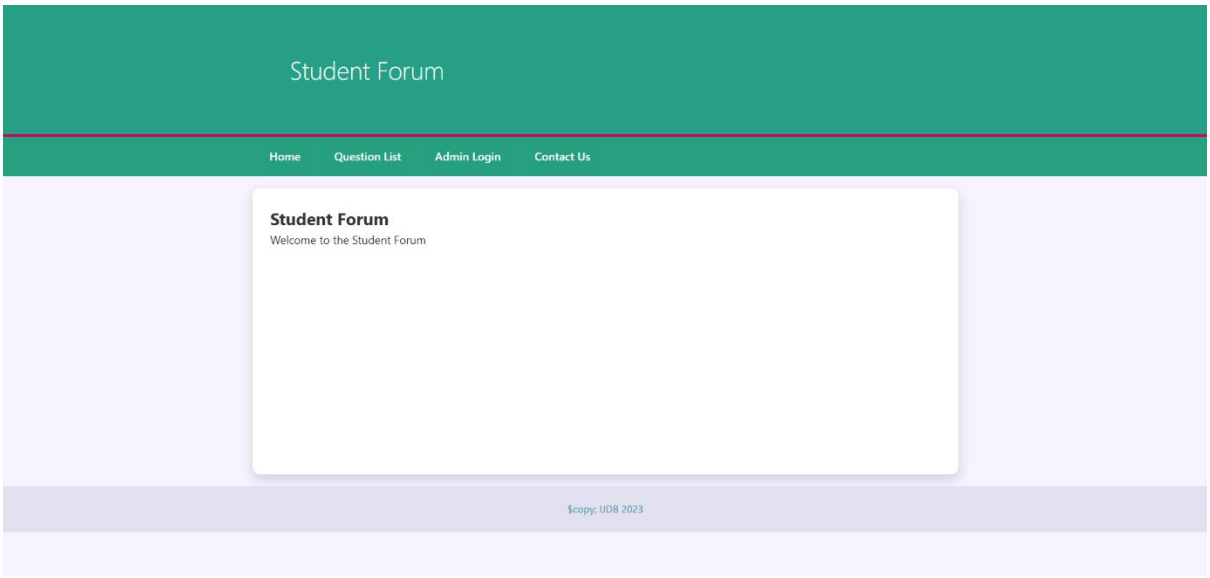
My goal is to create a place where you can comfortably ask questions, share experiences and study together. I also pay special attention to information security, committed to complying with regulations such as GDPR to protect user privacy.

This report will analyze in detail the features and performance of the website. At the same time, I will evaluate its impact on students and suggest future development directions. Through this project, I hope to help students be more proactive in their studies and build a strong knowledge sharing community.

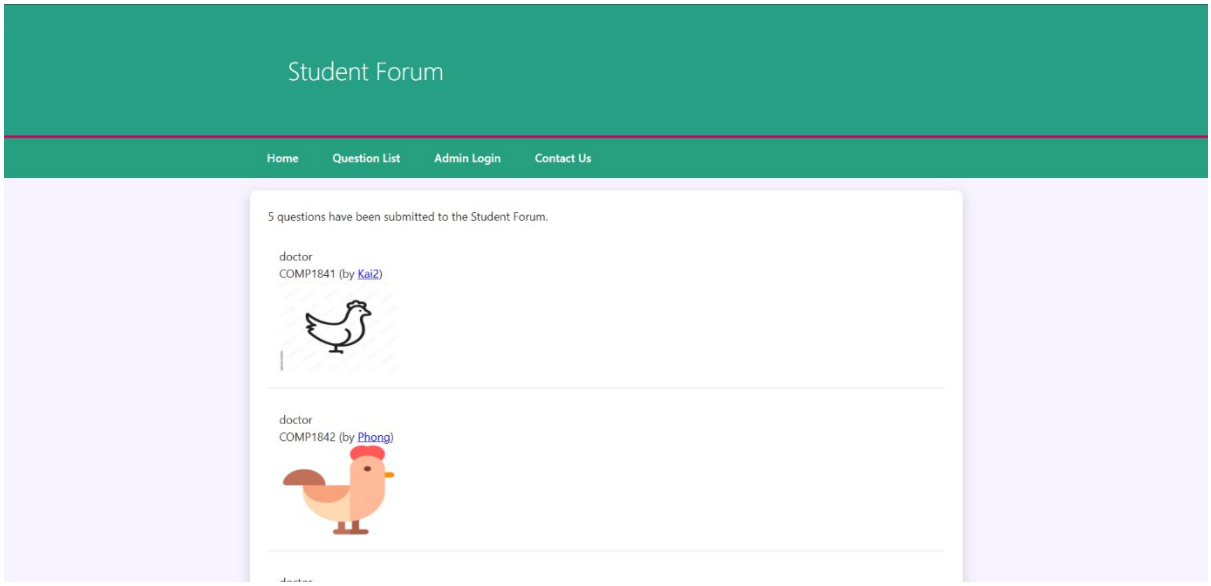
2. System Development

2.1. Design of the pages

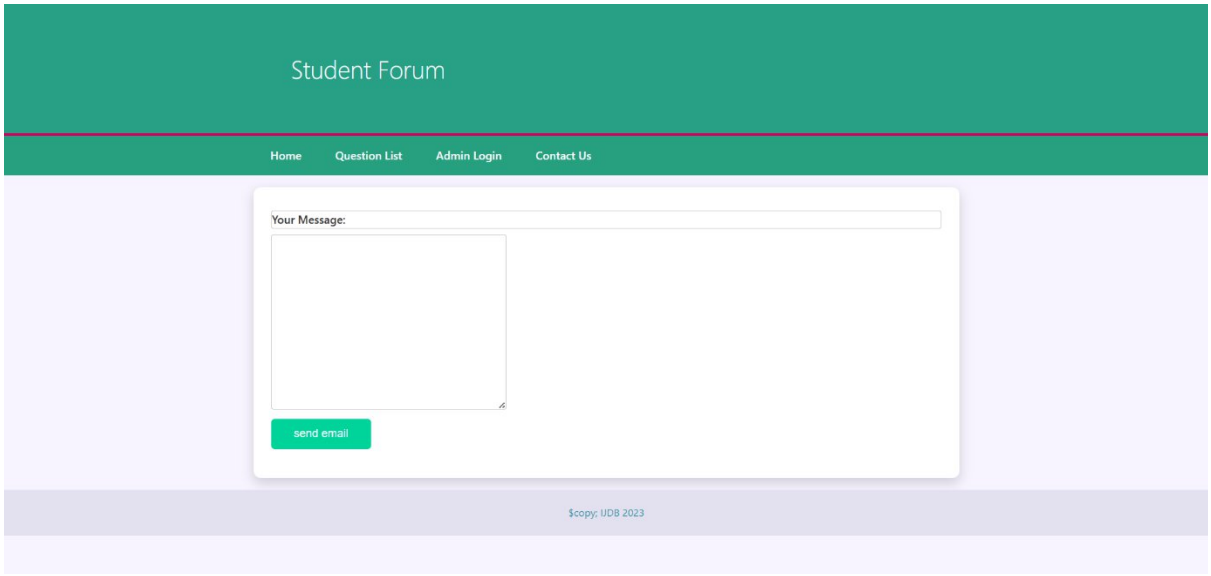
Public Site



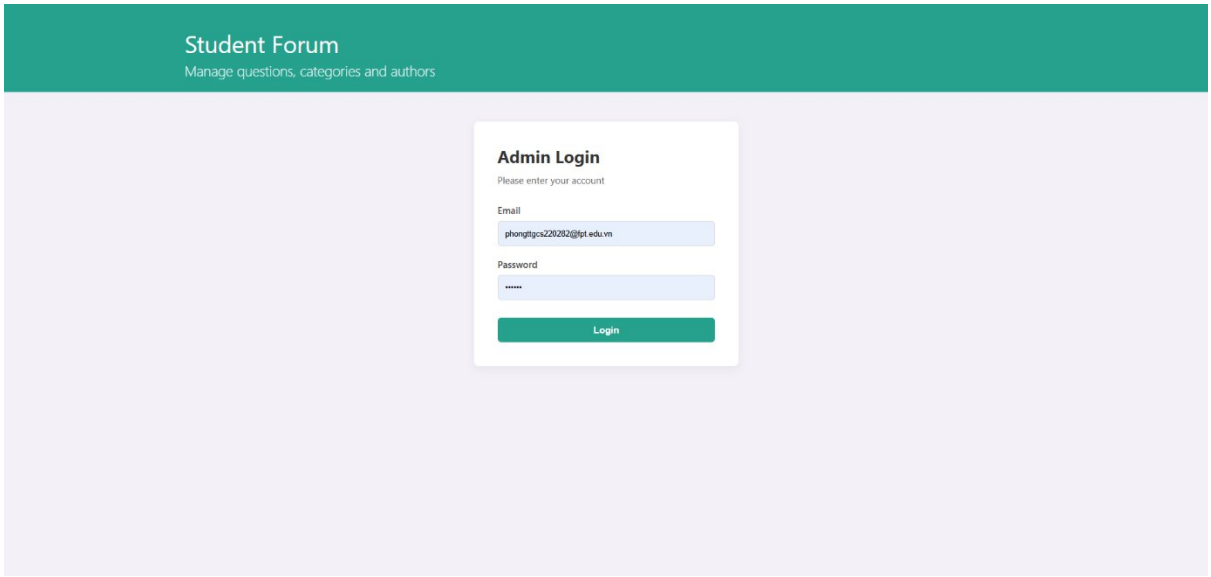
Home page



Question list page

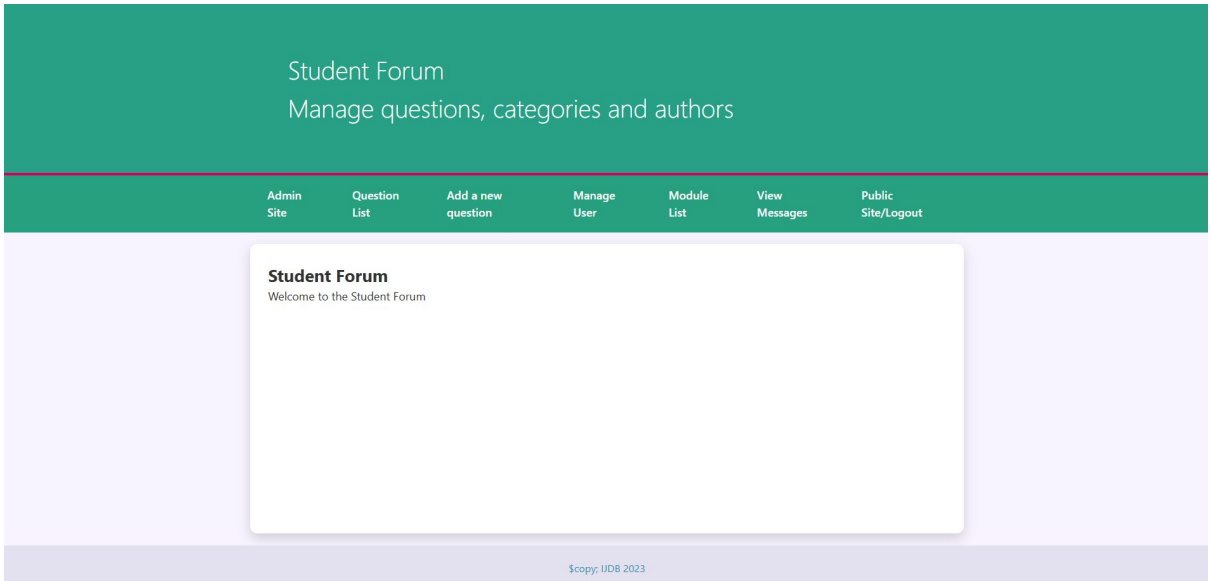


Contact us page

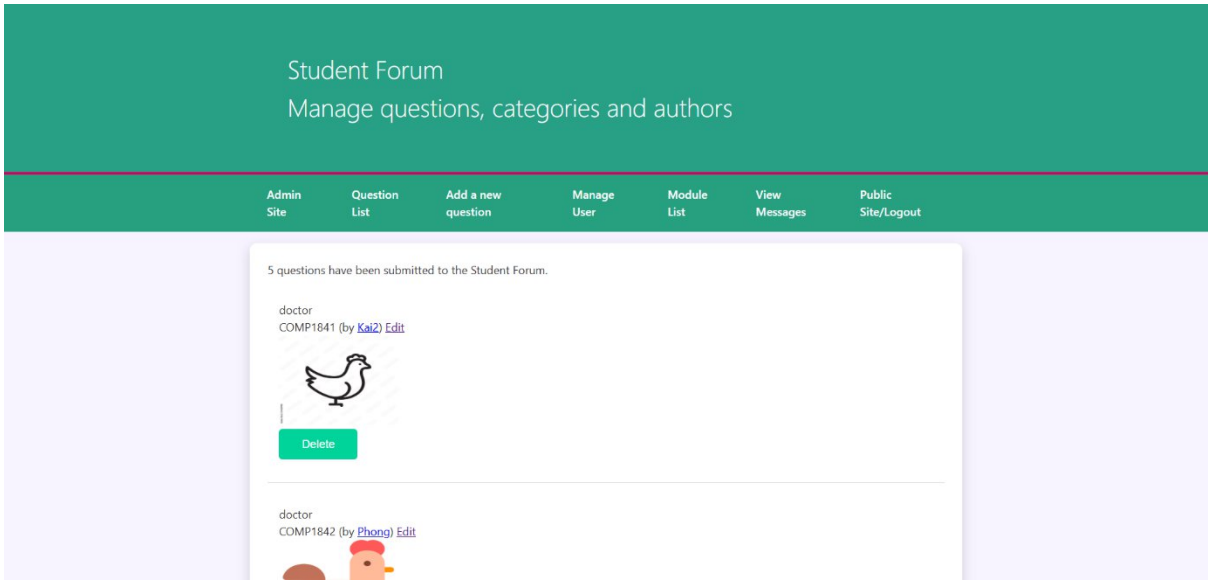


Login page(to Admin/User Web)

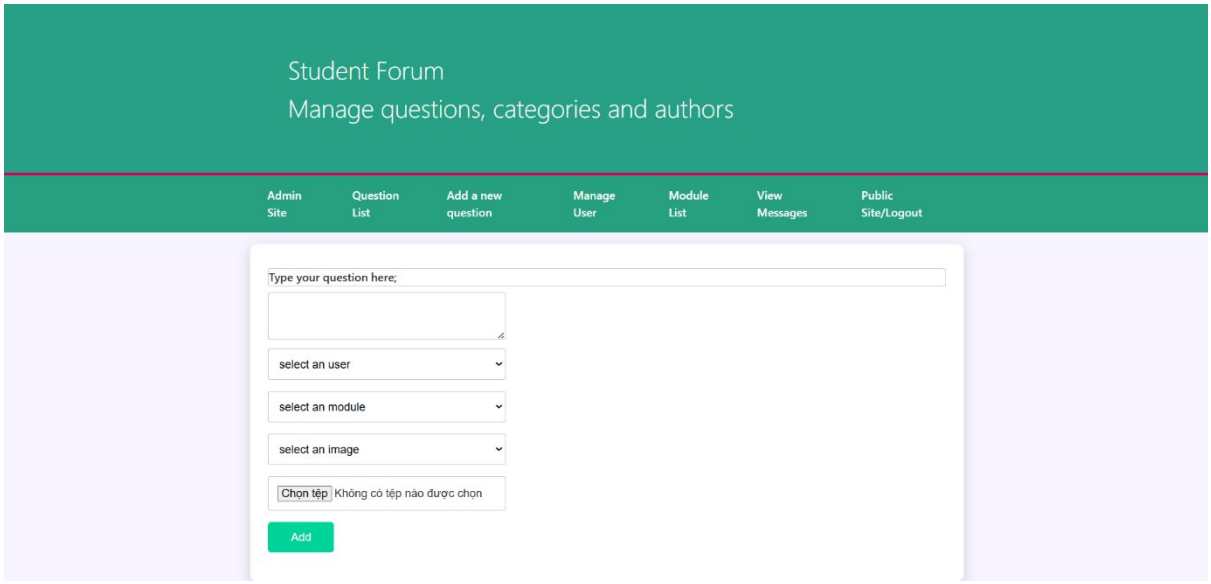
Admin/User Site



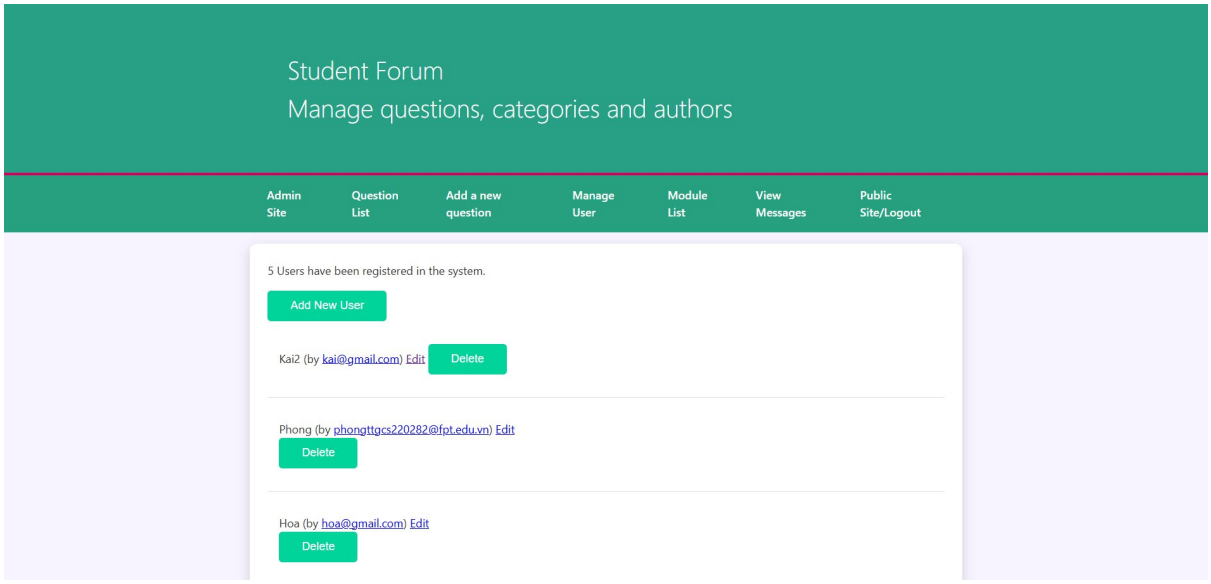
Home page



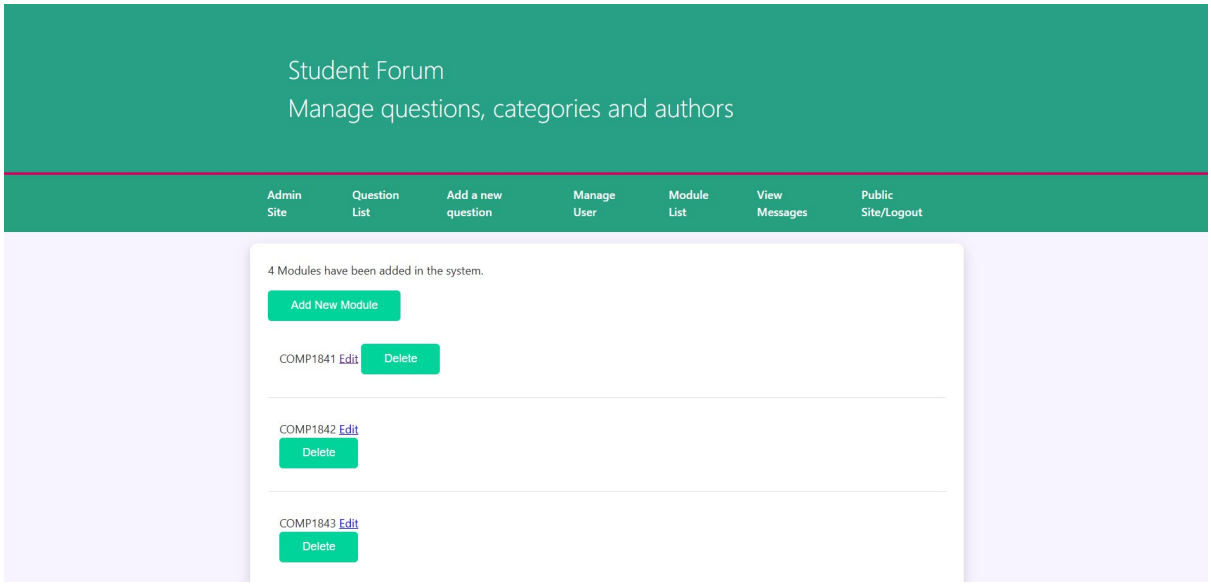
Question list page



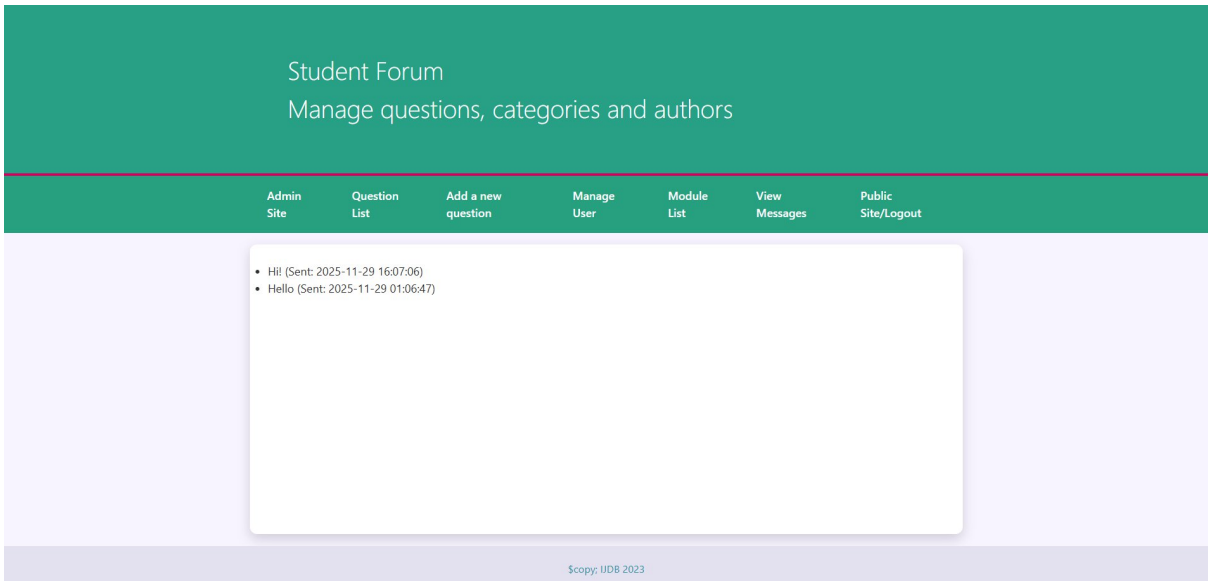
Add new question page



Manage user page



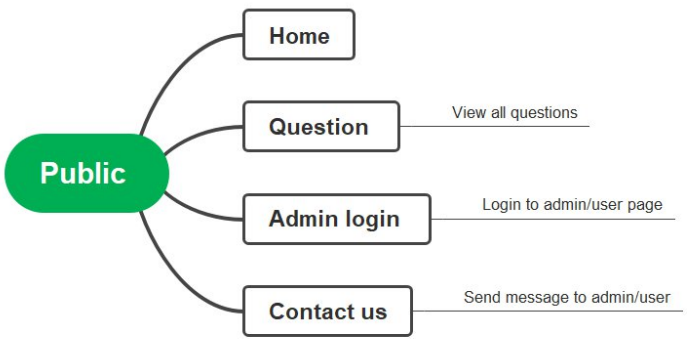
Module list page



View message page

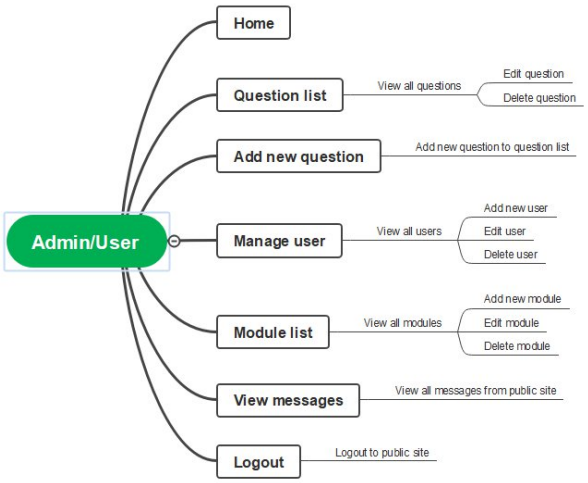
2.2. Navigation Structure

Public Site



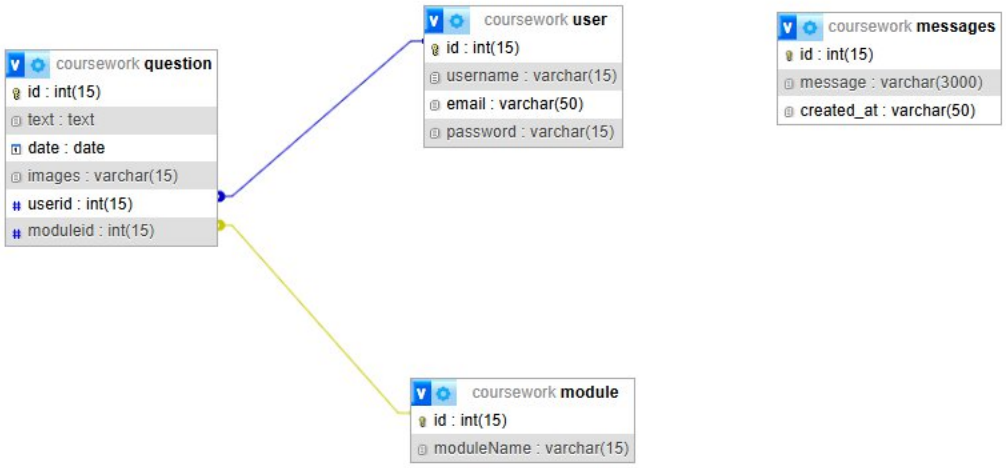
Public site(structure)

Admin/User Site



Admin/user site(structure)

2.3. Data diagrams



Data diagram(PHP MyAdmin)

3. Investigation of the technologies used and their justification

PHP:
PHP is a popular server-side scripting language that acts as the "brain" that controls the entire logic of this project.

Practical application:
In files such as adduser.php or edituser.php, PHP is responsible for receiving requests from users (when they click the "Save" button), handling validation logic (such as using if (isset($_POST['username'])) to confirm whether data has been sent or not) and navigating the website. In particular, PHP also helps protect the system from errors with the try...catch structure and manages the cache through the ob_start() function, helping the interface load more smoothly.

The Processing Engine:
PHP serves as the central engine of the website. Unlike a static page that simply displays fixed text, PHP enables the website to "think" and react. In this project (specifically in files like adduser.php), PHP captures the raw data typed by a user, processes logical checks (such as ensuring a username is not empty), and determines the next action. It transforms the website from a simple display into an interactive tool capable of handling complex tasks like user registration and login verification in real-time.

MySQL and PDO:
The project uses the MySQL database management system to store information persistently.

Practical application:
The Adminfunction.php file is the clearest demonstration of this technology. Instead of writing messy SQL code, the system uses the PDO (PHP Data Objects) library to connect and manipulate data. Functions such as updateUser or insertUser use the "Prepared Statements" technique (pre-prepared statements) with parameters such as :username, :email. This not only helps store and retrieve data (such as a list of users in the user table or messages in the messages table) but also absolutely prevents the risk of malicious code injection attacks (SQL Injection).

The Structured Data Archive:
For a forum to function, it needs a long-term memory. MySQL provides this by acting as a highly organized digital archive. Instead of storing data randomly, MySQL arranges information into strict tables—similar to advanced spreadsheets—linked by unique IDs. This structure allows the system to instantly locate a specific student's profile or a specific forum question among thousands of records. This ensures data integrity and fast retrieval speeds, which are essential for a smooth user experience.

The Secure Connection Layer:
Connecting the processing engine (PHP) to the data archive (MySQL) creates a potential security risk. To solve this, I utilized PDO (PHP Data Objects) as a secure interface layer. PDO uses a feature called "Prepared Statements," which separates the code instructions from the user's data. This ensures that any input entered by a user is treated strictly as text, not as executable code. This effectively neutralizes SQL Injection attacks, keeping the database safe from unauthorized manipulation.

HTML/CSS:
These are standard technologies for building the "framework" and "paint" for the website interface.

Practical application:
In the project, we apply the separation model: The processing logic is separate, and the interface is in template files with the extension .html.php (e.g. adduser.html.php, mailform.html.php). HTML creates input fields and buttons, while CSS (like the previously mentioned login.css file) uses Flexbox to align the layout beautifully, creating a modern and intuitive user experience.

Security Mechanism:
Although not a separate language, security techniques are tightly integrated into the code.

Practical application:
In the contact.php file and the display files (view_contact_send.php), the htmlspecialchars function is thoroughly used. This technology ensures that any message or user data entered is converted to plain text before being displayed. This helps prevent malicious scripts that can break the interface or steal administrator information.

Reason for the choice:
The choice of these technologies is based on stability, popularity and high controllability. PHP and MySQL are the classic "pair" that allows building a powerful server-side system without complicated installation. Separating HTML from processing logic makes the source code neat and easy to maintain. In particular, using PDO instead of the old MySQL connection shows that the project prioritizes security and modernity. Overall, this is a solid platform, easy to extend and suitable for both learning and practical implementation purposes.

4. Legal, social, ethical, personal data storage, and GDPR rules

1. Legal:

- Personal Data Protection (Replacement of EU GDPR):
- In Vietnam, Decree 13/2023/ND-CP is considered the "Vietnamese version of GDPR". For this project, that means:
- Consent: We cannot collect information arbitrarily. When students register, the system must clearly inform them what they will provide (e.g., email and name only) and what that data will be used for (login, not sold to third parties).
- Right to withdraw: Students have the right to say "no" at any time. If they want to leave the forum, we must have a mechanism (like a delete account button) to remove all their data from the system immediately.
- Copyright and Intellectual Property (Intellectual Property Law):
- The forum is a place to share, but not a place to consume pirated documents.
- We need to respect the efforts of textbook and lecture authors.
- The forum's regulations will strictly prohibit the posting of pirated scans of books or documents that are not allowed to be shared. If detected, the content will be removed to protect the legality of the platform.
- Cyber Security (Cyber Security Law 2018):
- The server system and database must be located in Vietnam (if the scale requires it) and more importantly, there must be measures to protect against cyber attacks to steal user data.

2. Social & Ethical:

- Academic Integrity:
- A question and answer forum can easily become a place to "ask for help with homework".
- Problem: Students copy each other's work or ask others to do the exercises.

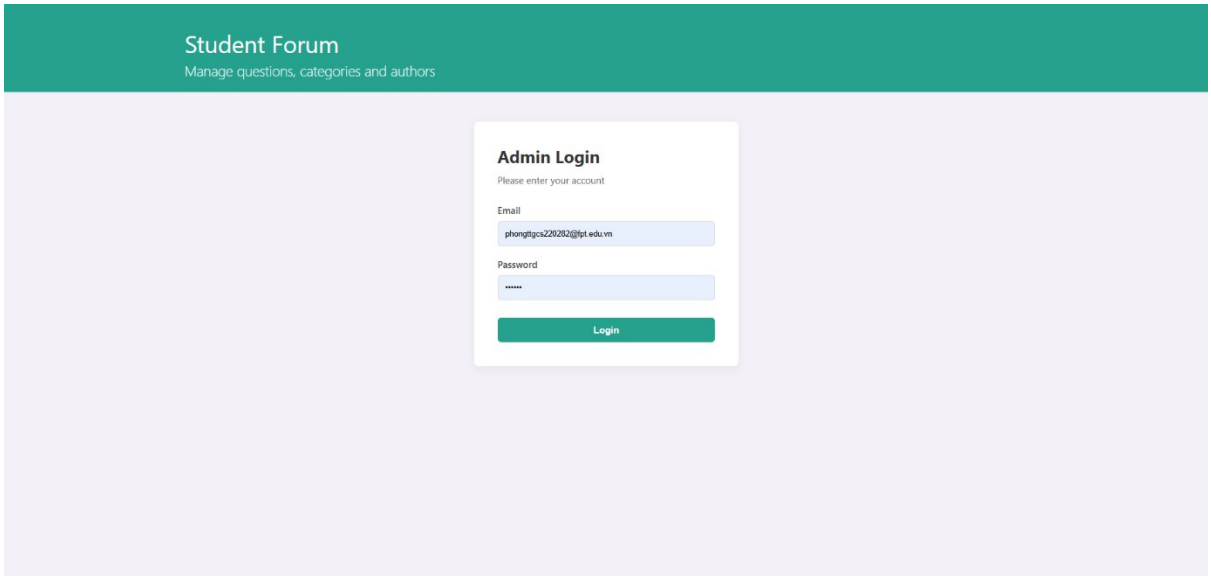
- Solution: Orient the forum as a place to discuss methods and suggest solutions, not a place to provide ready-made answers (copy-paste). We encourage a culture of citing sources when providing information.
- Anti-Cyberbullying:
- Virtual network but real damage. The forum needs to have a moderation mechanism (filtering bad keywords, reporting violations) to prevent hate speech, bullying or harassment between students. Everyone, whether good or bad, has the right to be respected when asking questions.
- Inclusiveness and Accessibility:
- No one should be left behind. The web interface needs to be designed clearly enough, with good contrast so that students with poor eyesight or difficulty in operation can still use it. This is not only ethical but also fair in education.

3. Personal Data Storage and GDPR Compliance:

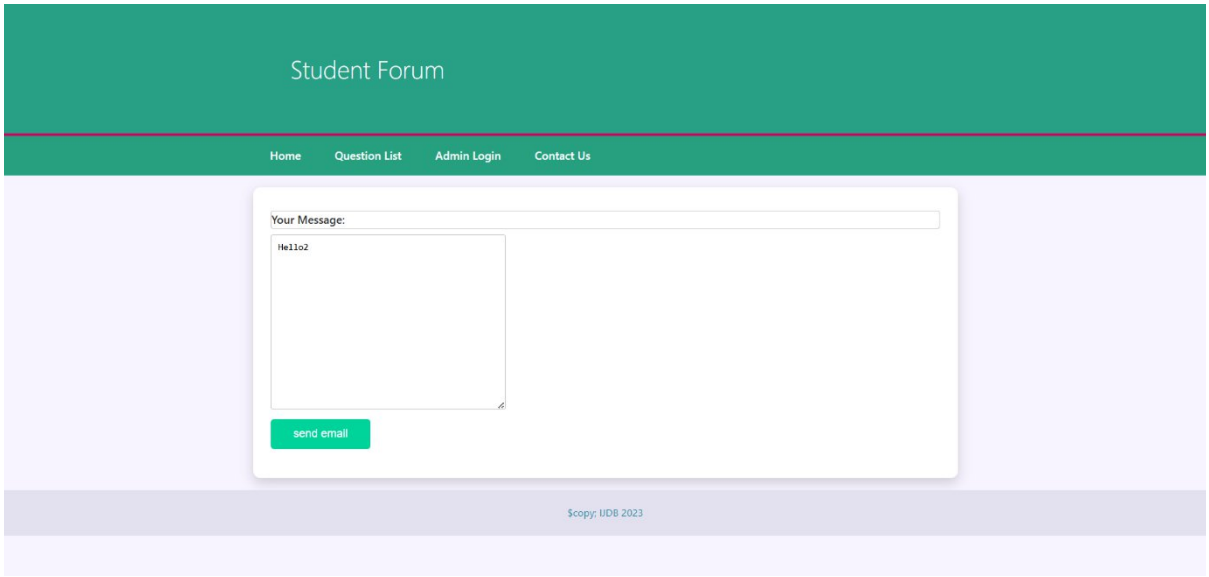
- "Data Minimization" Strategy:
- The golden rule is: "Don't keep what you don't need." The system only requires the minimum information for an account to function (Username, Email). We do not collect phone numbers, home addresses or dates of birth if the web function does not really need it. Keeping less data means reducing risks if there is an incident.
- Secure Storage:
- Data should never be left in plain text.
- User passwords must be hashed before being stored in the database. Even if the Admin or Hacker accesses the Database, they cannot know what the user's real password is.
- Use secure connections (like PDO in PHP) to prevent hackers from injecting malicious code into the database.
- Data Retention Policy:
- Old data, deleted accounts, or outdated logs should be periodically purged. This not only helps the system run faster, but also ensures that we don't store users' personal information longer than necessary.
- User Access:
- Put the power back in the hands of users. They should be able to review their personal information, make corrections if they make mistakes, and decide to delete their account without having to send permission emails or wait for complicated approvals.

5. System overview

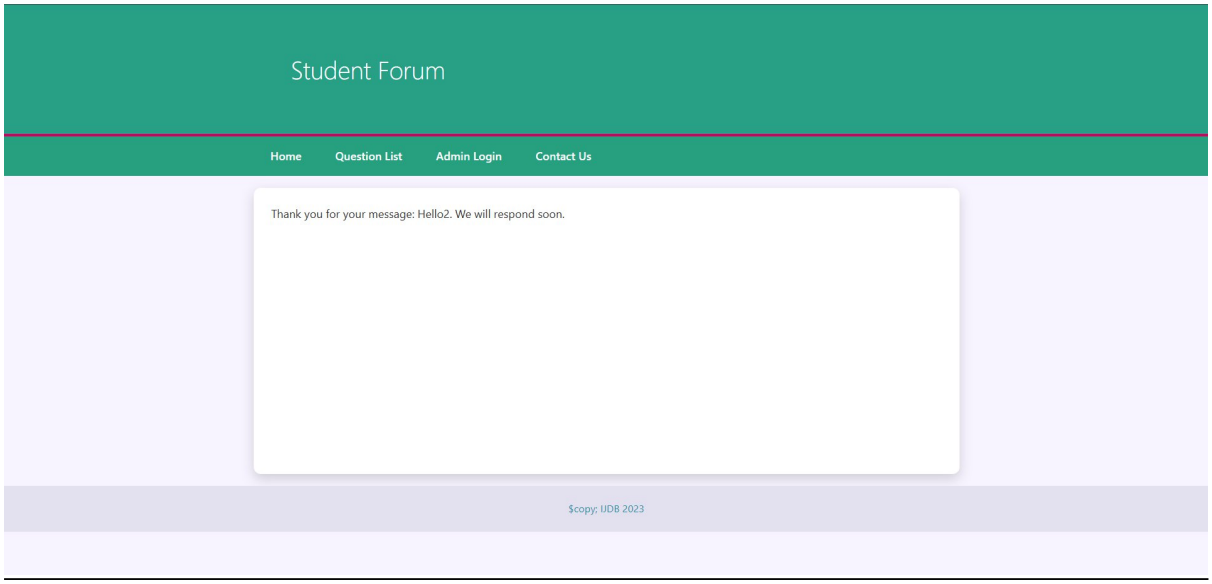
Public Site



Login(log into admin/user site)

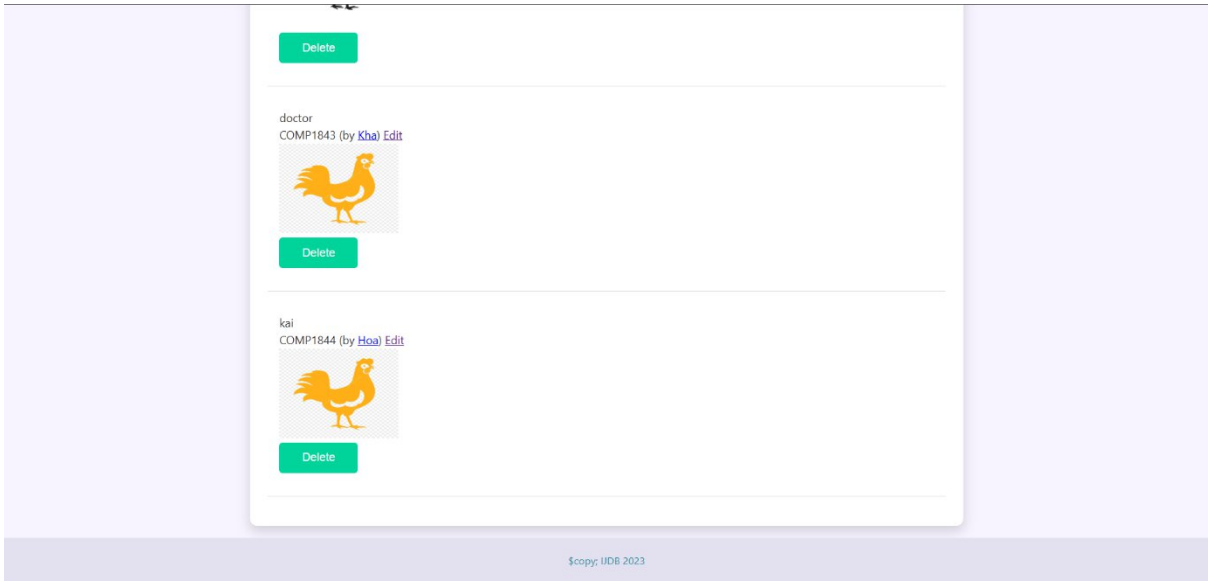


Contact us(send message to admin/user)

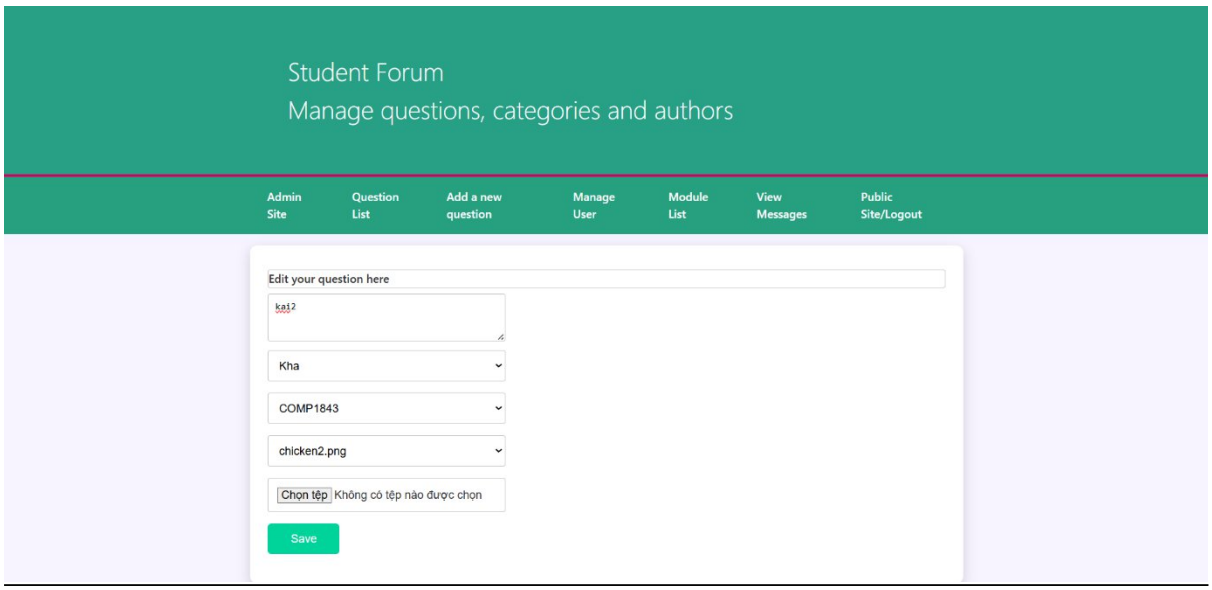


Contact us(send message complete)

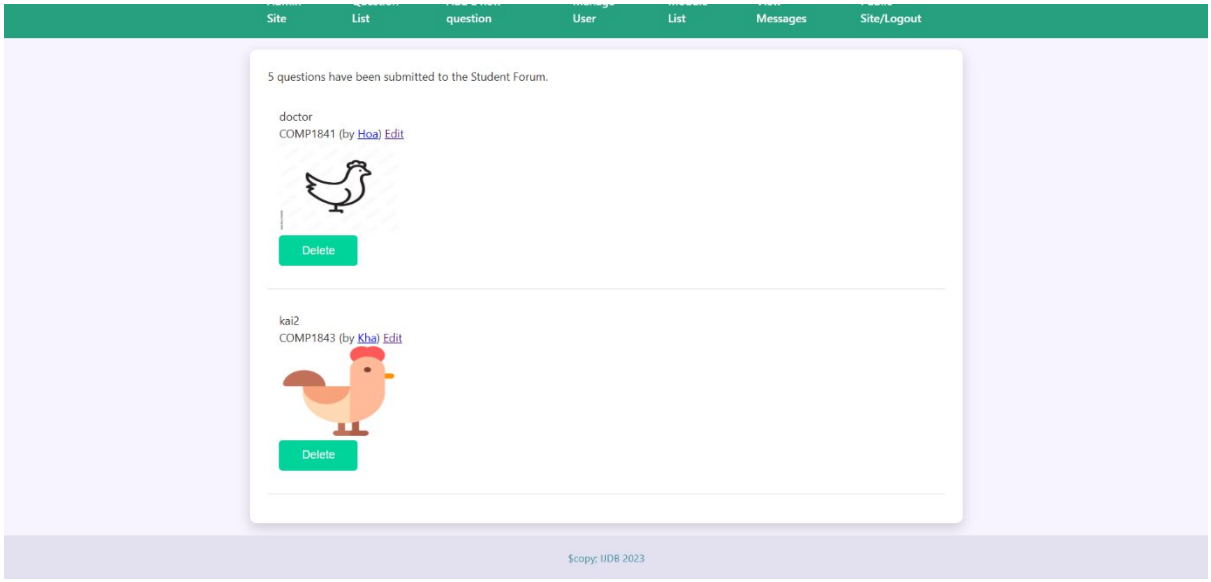
Admin/User Site



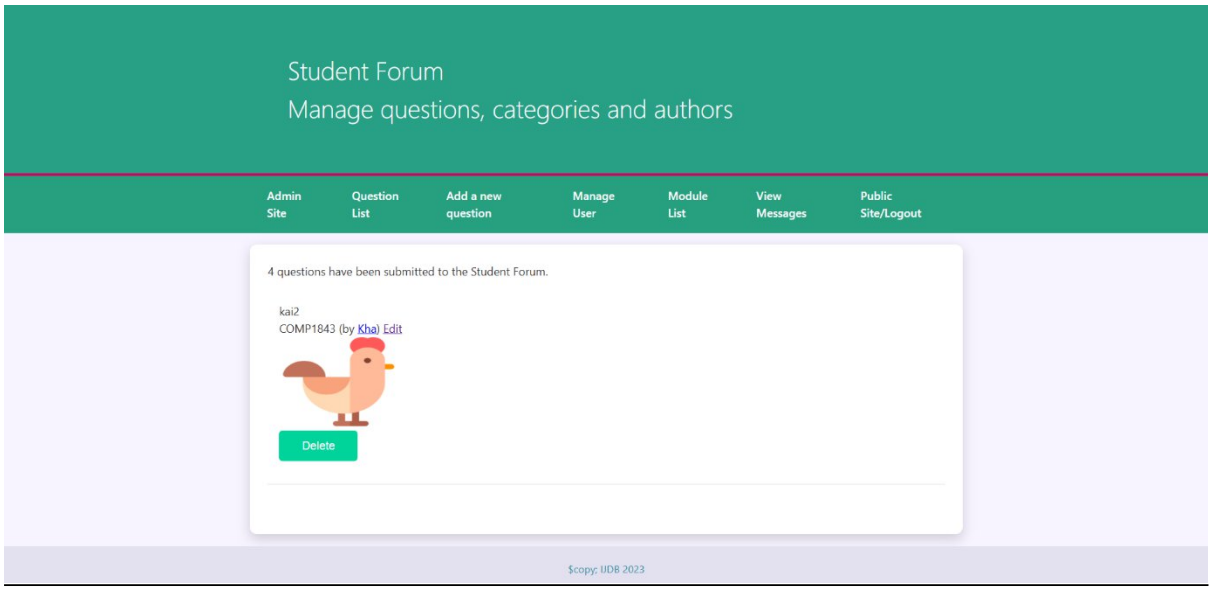
Before edit question



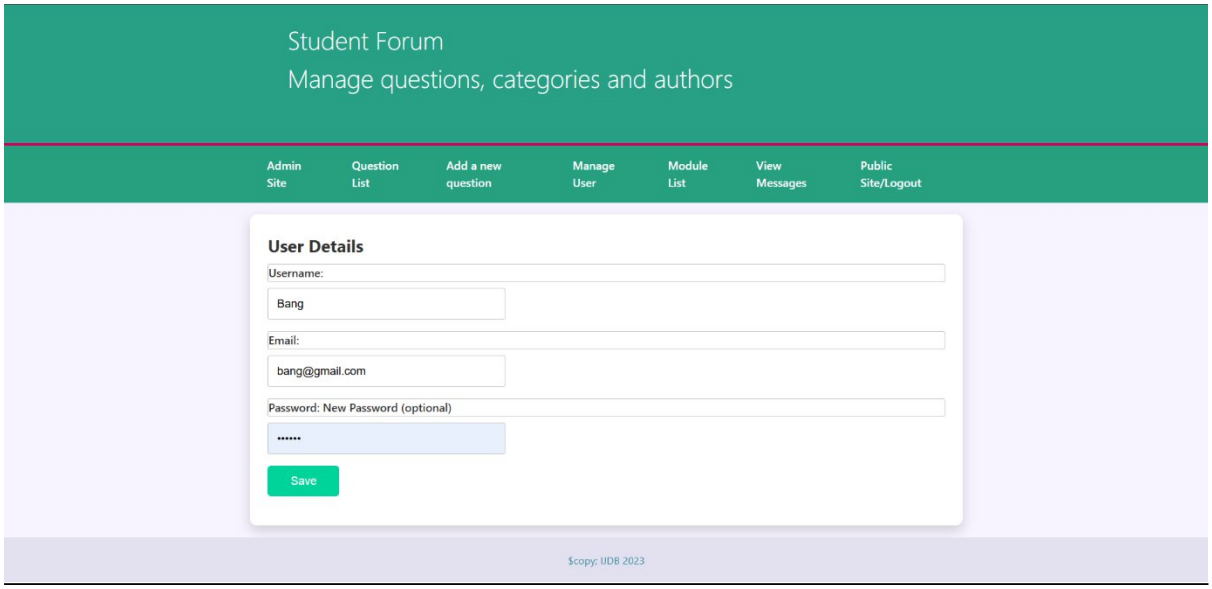
Edit question



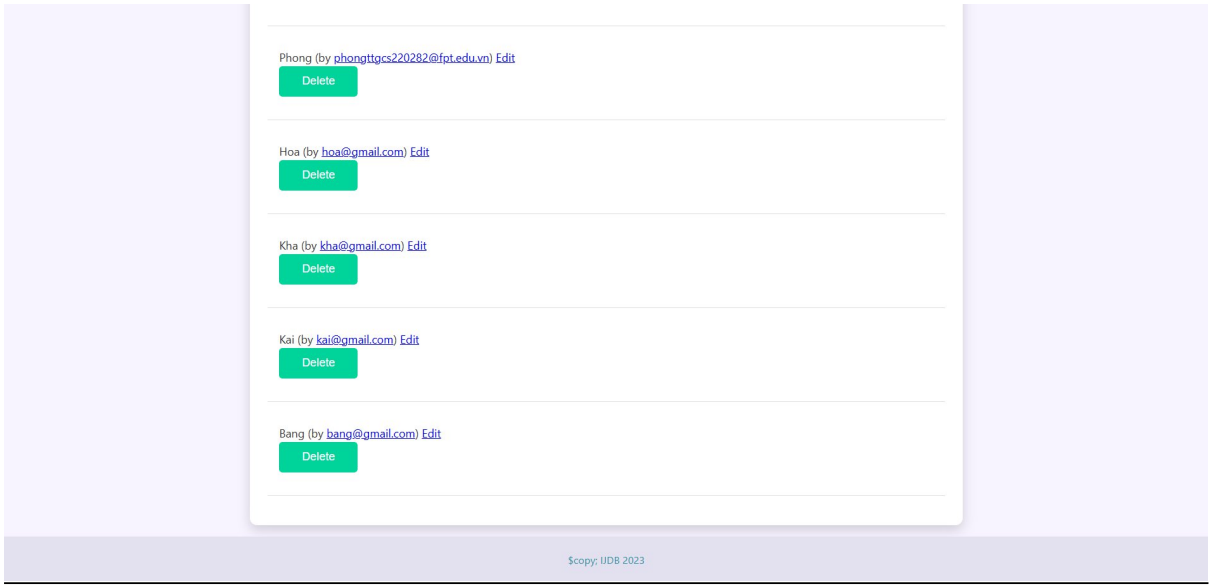
After edit question



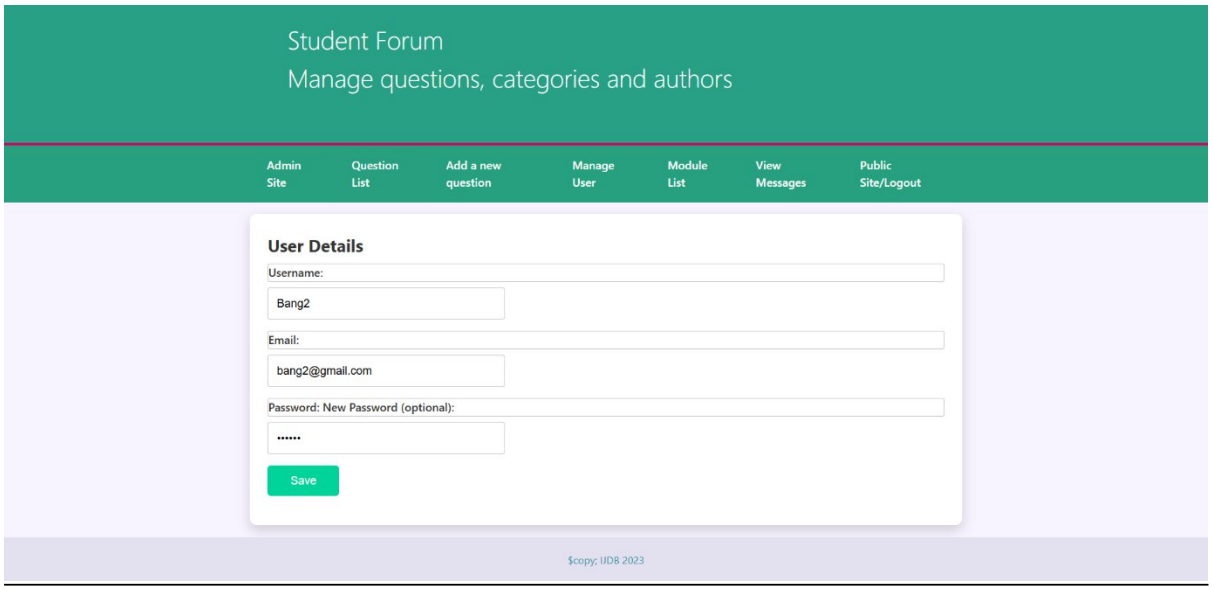
Delete question



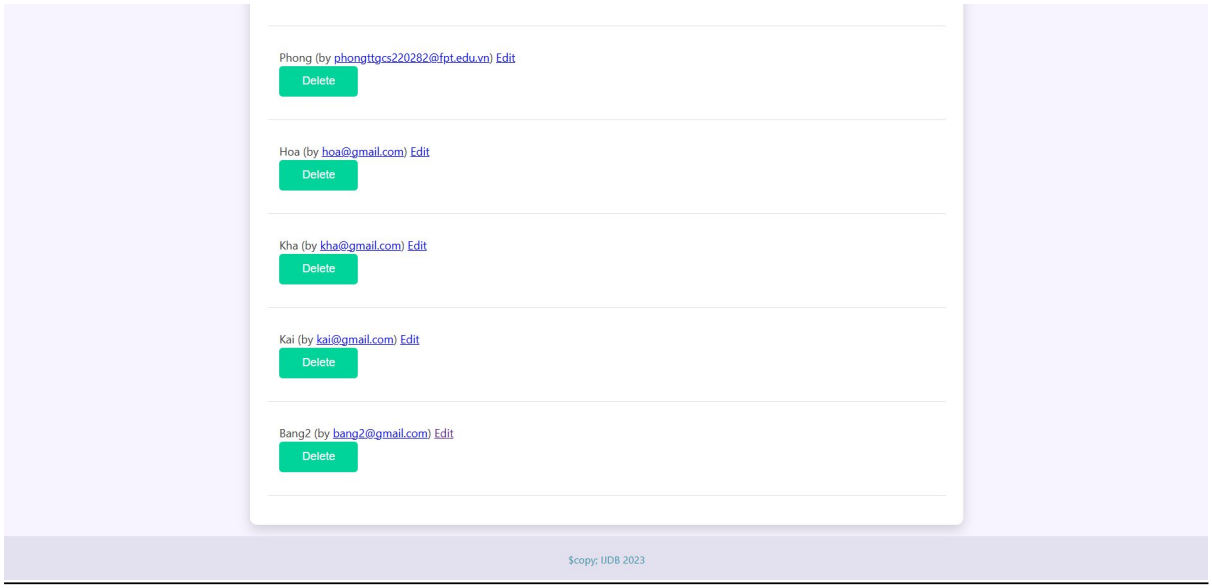
Add user



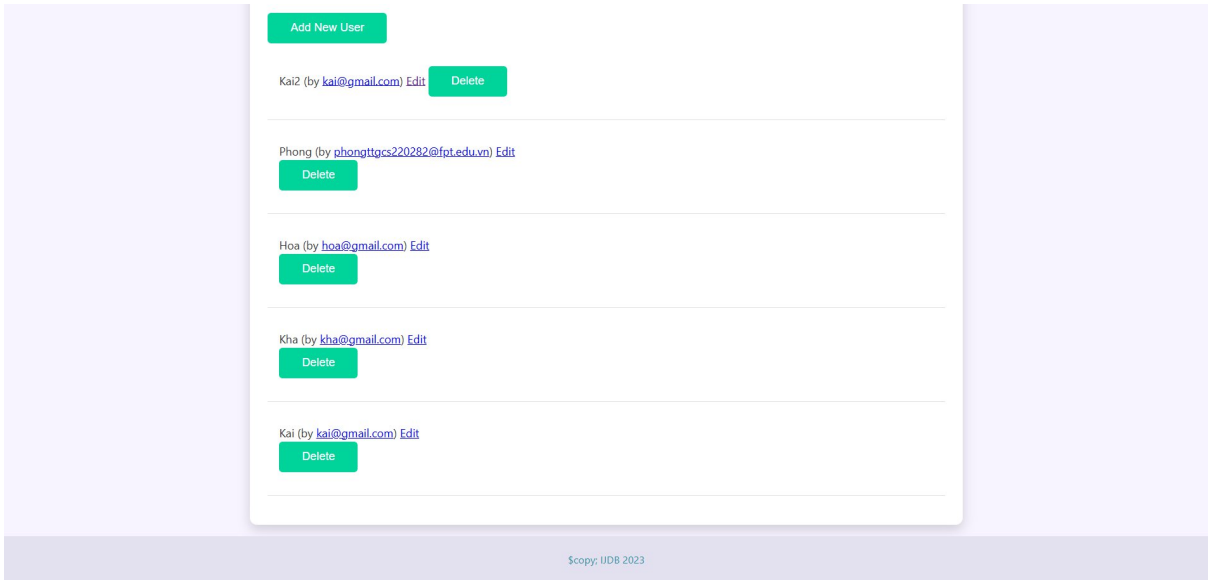
After add user



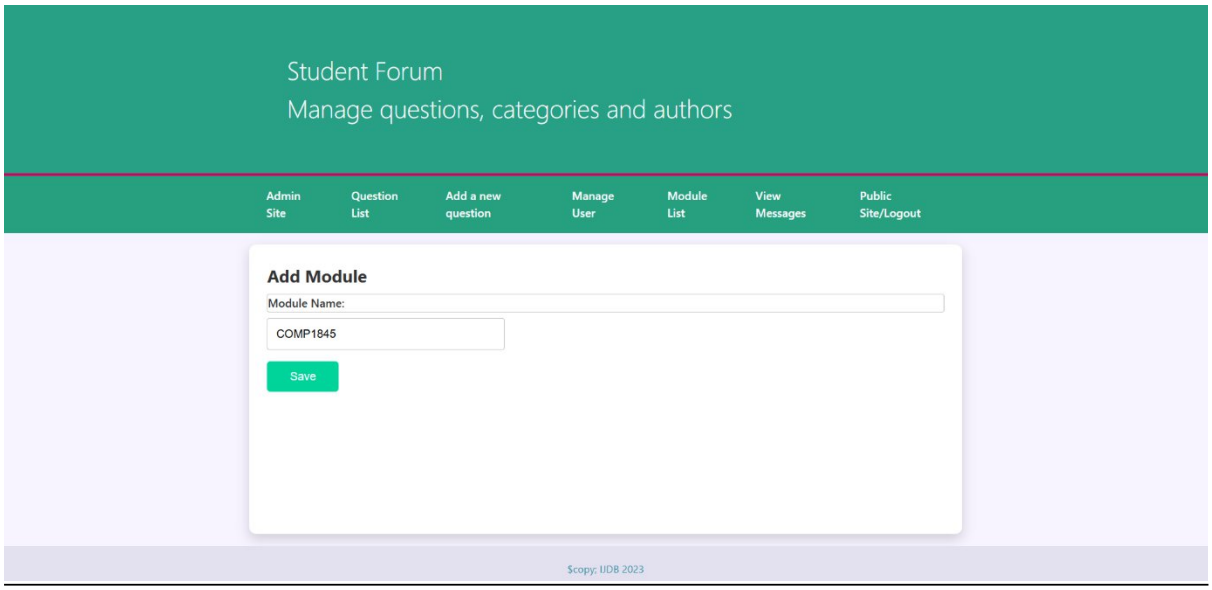
Edit user



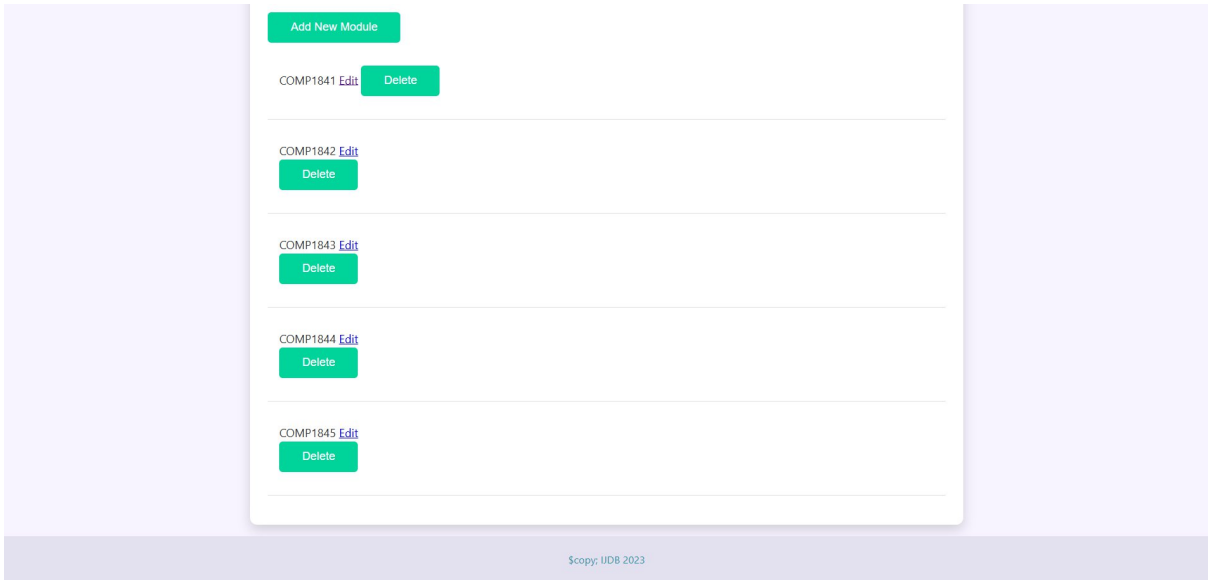
After edit user



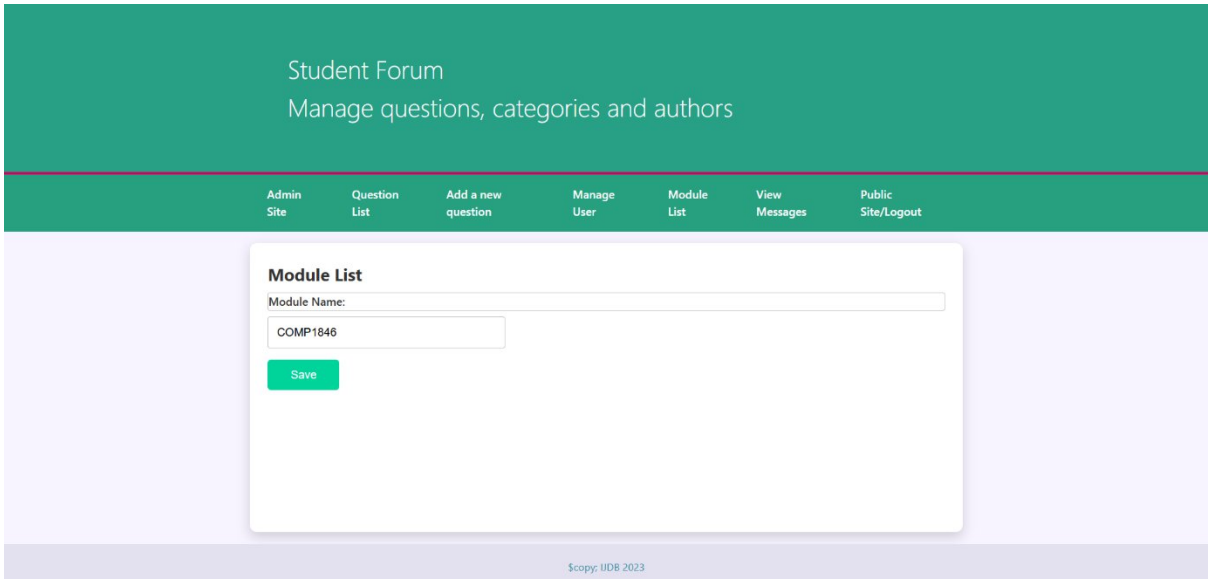
Delete user



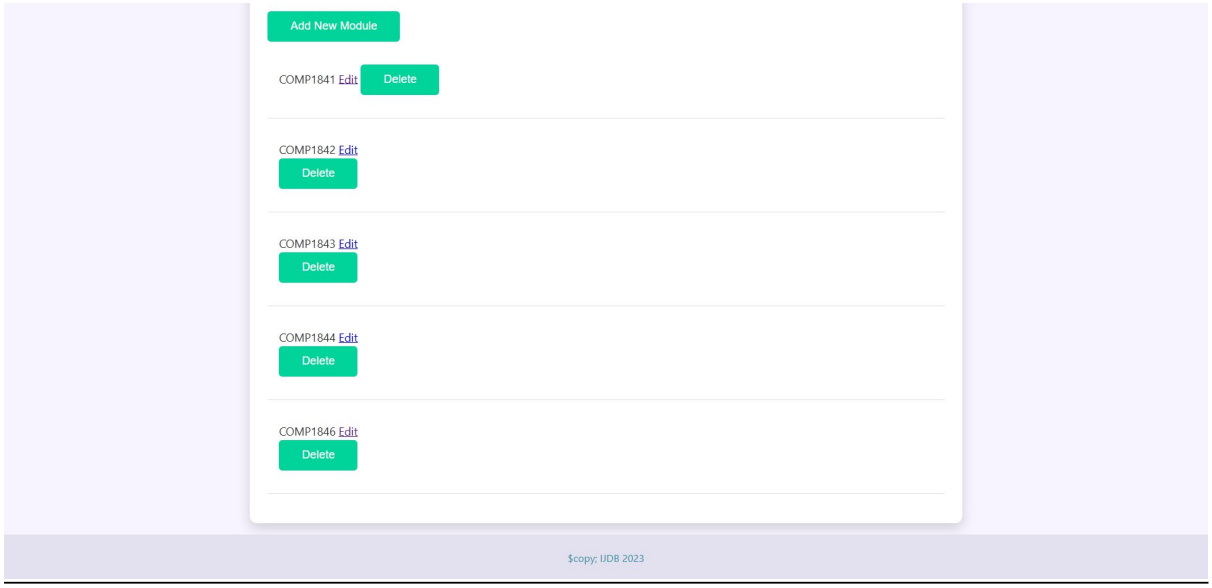
Add module



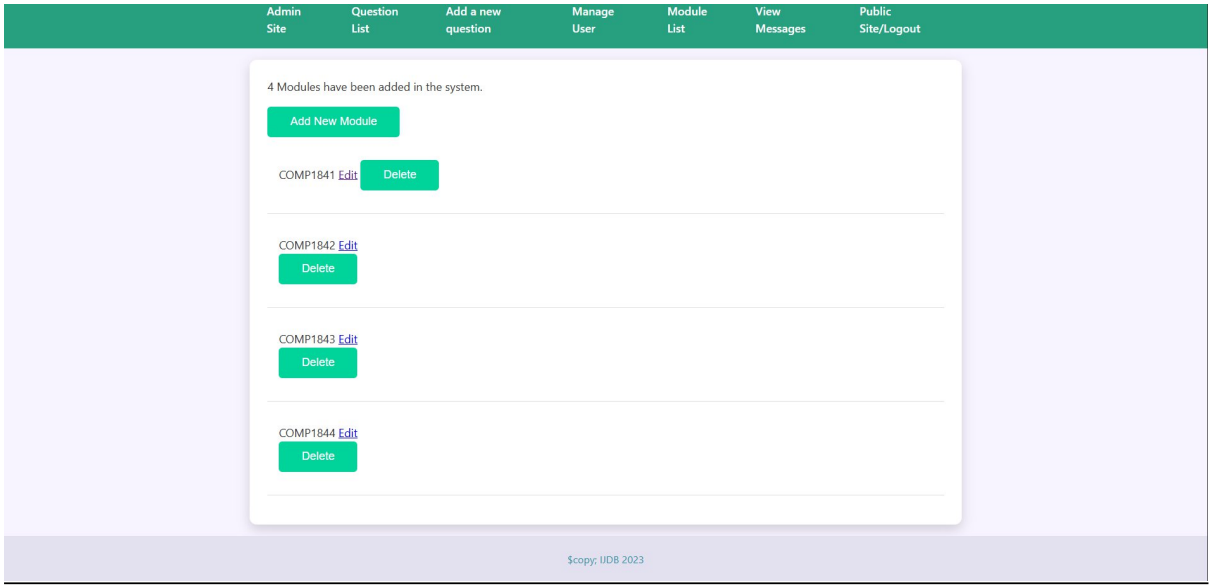
After add module



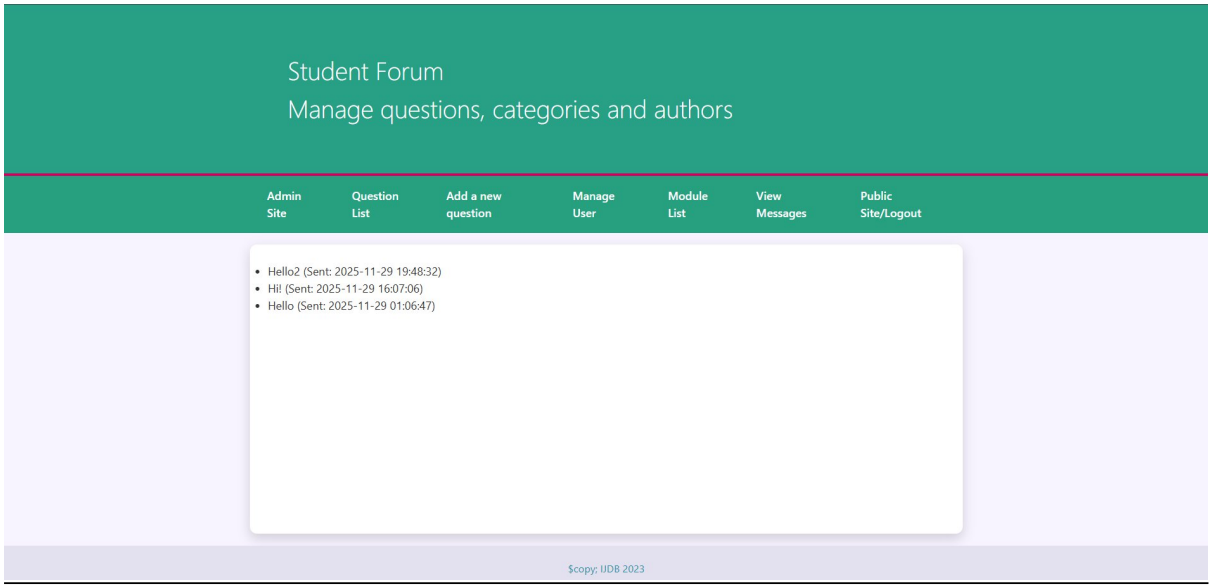
Edit module



After edit module

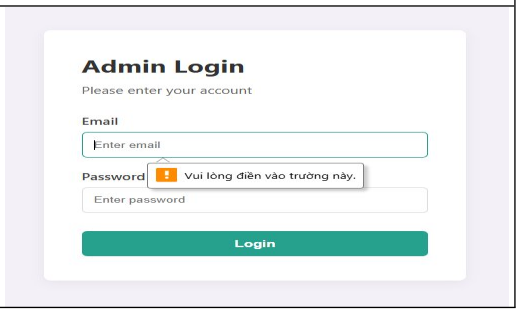


Delete module



View message(from public site)

6. Testing Schedule

Test case	Test content	Test data	Expected	Result
1	Incorrect account or password	phongttgcs220282 pass: hecret	Login to admin/user site	sorry wrong username or password go to login
2	Blank account or password	Account blank Password blank	Login to admin/user site	

7. Conclusions and future recommendations

1. Conclusions

Based on the results of source code implementation and system testing, I draw the following conclusions about the effectiveness of the project:

Functional stability: The project has successfully built the core of the admin page (Admin Dashboard). The most important functions such as adding, editing, deleting users (users.php) and receiving feedback (contact.php) have worked correctly, the data flow from the Form to the Database is smooth.

Source code structure: The clear separation between the logic processing file (such as adduser.php) and the interface file (such as templates/adduser.html.php) is a great success. This helps the code to be neat, easy to debug and meets modern web programming standards.

Data security: Applying PDO technology instead of old connection methods has helped the system achieve basic security standards, preventing SQL Injection errors when processing input data.

2. Future Recommendations

To develop from a platform project to a complete product for widespread use by students, I propose the following improvements:

Upgrading Password Security: Currently the system focuses on operational functions. The next step is to integrate strong encryption algorithms (such as Hashing) for user passwords to ensure absolute security even if data is exposed.

Optimizing Interface (UI/UX): The current interface is purely functional. It is necessary to integrate a design framework (such as Bootstrap) so that input forms and user lists display more beautifully and intuitively on mobile phones.

Automating Moderation: Instead of administrators having to manually view the message list in view_contact_send.php, the system should have an automatic keyword filter to immediately block spam or inappropriate language.

Expanded Search: When the number of students participating is large, manually searching the list is not feasible. It is necessary to add a search toolbar and advanced data filtering for more efficient management.

8. Harvard Referencing

PHP.net (2024) PHP: Hypertext Preprocessor Manual. Available at: https://www.php.net/manual/en/ (Accessed: 29 November 2025).

Oracle (2024) MySQL Reference Manual. Available at: https://dev.mysql.com/doc/refman/8.0/en/ (Accessed: 29 November 2025).

OWASP (2024) SQL Injection Prevention Cheat Sheet. Available at: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (Accessed: 29 November 2025).

W3C (2023) Web Content Accessibility Guidelines (WCAG) 2.2. Available at: https://www.w3.org/TR/WCAG22/ (Accessed: 29 November 2025).