VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**Pham Vu Quoc Cuong – 517H0040**

**Tran Pham Anh Tuan – 517H0093**

**WEBSITE MANAGE WORK PROCESSES YEARLY FOR FACULTY OF IT**

**INFORMATION TECHNOLOGY PROJECT 2**

**SOFTWARE TECHNOLOGY**

**HO CHI MINH CITY, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**Pham Vu Quoc Cuong – 517H0040**

**Tran Pham Anh Tuan – 517H0093**

**WEBSITE MANAGE WORK PROCESSES YEARLY FOR FACULTY OF IT**

**INFORMATION TECHNOLOGY PROJECT 2**

**SOFTWARE TECHNOLOGY**

Supervising teacher

**MSc. Doan Xuan Thanh**

**HO CHI MINH CITY, 2024**

**ACKNOWLEDGEMENTS**

As we reflect on the journey of completing Information Technology Project 2, we must acknowledge that our individual endeavors were significantly bolstered by the collective wisdom and support of our esteemed tutors and mentors.

Foremost, we extend our profound appreciation and highest regards to the distinguished faculty and dedicated staff of the Information Technology Department at Ton Duc Thang University. Their unwavering commitment to academic excellence has profoundly influenced our scholarly pursuits.

Particularly, we reserve our most heartfelt gratitude for our mentor, Professor Doan Xuan Thanh. His meticulous guidance and invaluable teachings have been the compass that steered us through the complexities of this project.

Conscious of our academic infancy and the limitations, we concede that our project may harbor inadvertent errors. It is with earnest humility that we invite the sagacious critique of our professors. Your insightful feedback is not only coveted for the refinement of this project but also as a cornerstone for our ongoing intellectual maturation.

In closing, we express our deepest and most sincere gratitude to all who have illuminated our path towards this academic milestone.

*Ho Chi Minh City, date 17 month 03 year 2024*

*Author*

*Cuong*

*Pham Vu Quoc Cuong*

*Tuan*

*Tran Pham Anh Tuan*

**THE WORK IS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

The work has been completed at Ton Duc Thang University. I hereby declare that this is my own research work and has been conducted under the scientific guidance of Professor Doan Xuan Thanh. The research content and results presented in this topic are honest and have not been published in any form before. The data in the tables used for analysis, comments, and evaluations were collected by the author from various sources, which are clearly cited in the reference section.

Furthermore, the project also utilizes some comments, evaluations, and data from other authors and organizations, all of which are properly cited and referenced.

Should any fraud be detected, I take full responsibility for the content of my project. Ton Duc Thang University is not related to any copyright or intellectual property violations that I may have caused during the execution (if any).

*Ho Chi Minh City, date 17 month 03 year 2024*

*Author*

*Cuong*

*Pham Vu Quoc Cuong*

*Tuan*

*Tran Pham Anh Tua*

**WEBSITE MANAGE EVENT PROCESSES YEARLY**

**FOR FACULTY OF IT**

**ABSTRACT**

**Project Objectives:**

* Develop a website that allows teachers to create, manage, and announce their events easily and effectively for participants who register for these events.

**Timeline and Personnel:**

* The project duration is from November 30, 2023 to March 17, 2024.
* Personnel involved in the project: Pham Vu Quoc Cuong, Tran Pham Anh Tuan.

**Technologies Used:**

* User Interface: ReactJs, Redux Tool, React-Big-Calendar.
* Backend: NodeJS, Node-Mailer, Node-Cron, ExcelJS, etc.
* Database: MongoDB.

**Key Features:**

* Login with Google account.
* Update account information, manage permissions, and set up initial login for new accounts.
* Create, edit, delete, and add personnel and student registrations for events and plans within a year.
* A UI for tracking the progress of activities and plans.
* Send email notifications to members added to event when a new event is created, when schedule of an activity changes, and report completion when an activity is finished.

**MỤC LỤC**

[LIST OF FIGURES vi](#_Toc161935010)

[LIST OF TABLE ix](#_Toc161935011)

[LIST OF ABBREVIATIONS x](#_Toc161935012)

[CHAPTER 1 : INTRODUCTION AND TOPIC OVERVIEW 1](#_Toc161935013)

[1.1 Reason for choosing the topic 1](#_Toc161935014)

[1.2 Target in implementing the topic 2](#_Toc161935015)

[CHAPTER 2: THEORETICAL BASIS 3](#_Toc161935016)

[2.1 Task Management System: 3](#_Toc161935017)

[2.2 Website design using ReactJS: 7](#_Toc161935018)

[2.3 MongoDB database: 8](#_Toc161935019)

[CHAPTER 3: RECOMMENDED MODEL 10](#_Toc161935020)

[3.1 Overview of the recommended model: 10](#_Toc161935021)

[3.2 Specific functions: 10](#_Toc161935022)

[3.3 Technology used: 10](#_Toc161935023)

[3.4 ERD model: 11](#_Toc161935024)

[CHAPTER 4: WORK SHARING. 17](#_Toc161935025)

[4.1 Work sharing Backend: 17](#_Toc161935026)

[4.2 Work sharing Fontend: 18](#_Toc161935027)

[4.3 Work sharing report: 18](#_Toc161935028)

[CHAPTER 5. EXPERIMENTATION 19](#_Toc161935029)

[5.1 Login Page: 19](#_Toc161935030)

[5.2 Account page: 22](#_Toc161935031)

[5.3 Home Page: 25](#_Toc161935032)

[5.4 Dashboard Page: 28](#_Toc161935033)

[5.5 User Page: 31](#_Toc161935034)

[5.6 Event page: 35](#_Toc161935035)

[CHAPTER 6. RECOMMENDATION AND CONCLUSION 40](#_Toc161935036)

[6.1 Conclusion: 40](#_Toc161935037)

[6.2 Future Directions: 41](#_Toc161935038)

[6.3 Summarize 41](#_Toc161935039)

[REFERENCES 42](#_Toc161935040)

# LIST OF FIGURES

[Figure 2.1 1: Task Management System. 3](#_Toc161876898)

[Figure 2.1 2**:** Some common features. 4](#_Toc161876899)

[Figure 2.1 3**:** Google Calendar. 5](#_Toc161876900)

[Figure 2.1 4: Notion. 5](#_Toc161876901)

[Figure 2.1 5: Trello. 6](#_Toc161876902)

[Figure 2.1 6: Microsoft teams. 6](#_Toc161876903)

[Figure 2.2 1: ReactJS. 7](#_Toc161876904)

[Figure 2.3 1: MongoDB. 9](#_Toc161876905)

[Figure 3.4 1: Relational model of student role. 11](#_Toc161876906)

[Figure 3.4 2: The relational model of teacher role. 12](#_Toc161876907)

[Figure 3.4 3: Relational model of the Admin role. 13](#_Toc161876908)

[Figure 3.4 4: Class Diagram. 14](#_Toc161876909)

[Figure 3.4 5: Sequence Diagram. 14](#_Toc161876910)

[Figure 3.4 6: Login Activity Diagram. 15](#_Toc161876911)

[Figure 3.4 7: Get User Info Actvity Diagram. 15](#_Toc161876912)

[Figure 3.4 8: Request Event Activity Diargram. 16](#_Toc161876913)

[Figure 3.4 9: Request Edit User Activity Diagram. 16](#_Toc161876914)

[Figure 5.1 1: UI page login. 19](#_Toc161932300)

[Figure 5.1 2: Sign in with google account. 20](#_Toc161932301)

[Figure 5.1 3: Backend code validates user 1. 21](#_Toc161932302)

[Figure 5.1 4: Backend code validates user 2. 21](#_Toc161932303)

[Figure 5.2 1: User profile tab. 22](#_Toc161876919)

[Figure 5.2 2: Join event history tab. 22](#_Toc161876920)

[Figure 5.2 3: The backend code retrieves user data. 23](#_Toc161876921)

[Figure 5.2 4: The backend code retrieves event history data. 23](#_Toc161876922)

[Figure 5.2 5: The backend code retrieves the number of events participated in the year and training points. 24](#_Toc161876923)

[Figure 5.3 1: Tab event on calendar. 25](#_Toc161876924)

[Figure 5.3 2: Event completeness tab 25](#_Toc161876925)

[Figure 5.3 3: Get all events in DB. 26](#_Toc161876926)

[Figure 5.3 4: The code filters events by userId if the role is student. 26](#_Toc161876927)

[Figure 5.3 5: The code filters events by userId if the role is teacher. 27](#_Toc161876928)

[Figure 5.4 1: Interface page of Dashboard. 28](#_Toc161876929)

[Figure 5.4 2: Code to get summary of event in year 29](#_Toc161876930)

[Figure 5.4 3: Code to get summary of event in month 29](#_Toc161876931)

[Figure 5.4 4: Code to get 10 newest events 30](#_Toc161876932)

[Figure 5.5 1: Interface of user page 31](#_Toc161876933)

[Figure 5.5 2: Searching user infomartion. 31](#_Toc161876934)

[Figure 5.5 3: Filter user. 32](#_Toc161876935)

[Figure 5.5 4: Edit user. 32](#_Toc161876936)

[Figure 5.5 5: Notification when user is successfully edited. 33](#_Toc161876937)

[Figure 5.5 6: Notification when user is successfully deleted. 33](#_Toc161876938)

[Figure 5.5 7: Code CRUD of user. 34](#_Toc161876939)

[Figure 5.6 1: Main interface of event management page. 35](#_Toc161876940)

[Figure 5.6 2: Event search function. 36](#_Toc161876941)

[Figure 5.6 3: Function to add new events. 36](#_Toc161876942)

[Figure 5.6 4: Email notification of event participation. 37](#_Toc161876943)

[Figure 5.6 5: View and edit event information. 37](#_Toc161876944)

[Figure 5.6 6: Email notification when editing event. 38](#_Toc161876945)

[Figure 5.6 7: Email when the user chooses to send a change email. 38](#_Toc161876946)

[Figure 5.6 8: Pop up confirm clears the event. 39](#_Toc161876947)

# LIST OF TABLE

[Table 4.1 1: Work sharing Backend. 17](#_Toc161876967)

[Table 4.2 1: Work sharing Fontend. 18](#_Toc161876971)

[Table 4.3 1:Work sharing Report 18](#_Toc161876975)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| IT | Information technology |

# CHAPTER 1 : INTRODUCTION AND TOPIC OVERVIEW

## Reason for choosing the topic

**a. Practicality:**

**Necessity**: The event management of the Faculty of Information Technology (IT) is currently facing many difficulties due to the lack of an effective management system, leading to information not being updated promptly, and difficulties in following up. Monitor, notify, and evaluate the effectiveness of events.

**Effective solution:** The event management website will help the IT faculty solve the above problems, bringing high efficiency in event management.

**Process automation:** The system will automate operations such as registering to participate, sending notifications, sending reports, etc., helping to save time and effort for the organizers.

**Update information instantly:** All information about the event will be quickly updated on the website, making it easy for participants to follow and track.

**Monitor and evaluate effectiveness:** The system provides detailed statistical reports on the status of events, registrants and event participants, etc., helping organizers evaluate event effectiveness and make improvements for next event.

**b. Workability:**

**Ability of extension:** The website can be expanded to manage many different types of events, from scientific conferences, symposiums to exchanges, workshops, etc.

**Flexibility:** The system can be customized to fit the IT faculty's unique needs and event management process.

**c. Benefits for students:**

**Provide complete and accurate information about events:** Students can easily search and follow information about IT faculty events.

**Reminder notification:** For students who participate in IT department events if events change.

**d. Benefits for IT faculty**:

**Improve event management efficiency:** The website helps the IT department manage events effectively, saving time and costs.

**Strengthen connections with students:** The website helps the IT faculty strengthen connection with students, creating conditions for students to participate in department activities.

**Conclusion**:

For the above reasons, the topic "Website manage event processes yearly for faculty of IT" is a practical, highly applicable and practical topic. Implementing this topic will bring many benefits to the IT department, teachers and students.

## Target in implementing the topic

Build an effective event management website, meeting the needs of managing and organizing events of the IT department.

**About function:**

Provides full event management functions, including: creating events, registering to participate, tracking progress, sending notifications, sending reports, ...

Support future system expansion of the IT faculty.

**About efficiency:**

Automate the event management process, helping to save time and effort for organizers.

Update information continuously and accurately, making it easy for participants to follow and grasp.

Improve event management efficiency, increase registration rates, and attract more attendees.

# CHAPTER 2: THEORETICAL BASIS

## 2.1 Task Management System:



Figure 2.1 1: Task Management System.

First, let's find out why building an event management website brings many of the above benefits compared to conventional management. Because the event management website is built as a task management software.

Task management software is a tool that helps users manage tasks, projects and jobs effectively.

**a. There are different types of task management software, including:**

* **Calendar application**: Helps you keep track of upcoming events, meetings, and tasks. Google Calendar is one of the popular examples.
* **To-do list app**: Helps you create a list of tasks to complete and track your progress. Notion is a popular example.
* **Project management application**: Helps you plan and manage complex projects. Asana and Trello are popular examples.
* **Collaboration applications**: Helps you work with others on your team on tasks and projects. Slack and Microsoft Teams are popular examples.

**b. Some common features of work management software:**



Figure 2.1 2**:** Some common features.

* **Create and manage tasks**: Allows you to create, edit, and delete tasks.
* **Arrange Tasks:** Allows you to arrange tasks by date, project, priority.
* **Monitor progress:** Allows you to track your progress on tasks and projects.
* **Remind:** Lets you set reminders for upcoming tasks.
* **Collaboration:** Allows you to work with others on tasks and projects.
* **Report:** Allows you to generate reports about your work performance.

**c. Benefits of using work management software:**

* **Increase efficiency**: Helps you get more done in less time.
* **Increase productivity**: Helps you focus on the most important tasks.

**d. Some popular task management software:**

**Google Calendar**: The popular calendar app lets you keep track of upcoming events, meetings, and tasks.

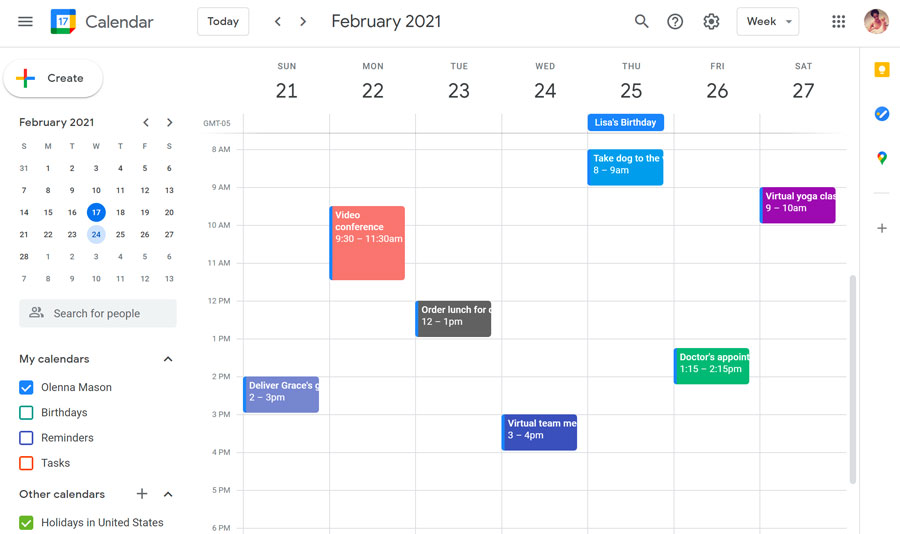


Figure 2.1 3**:** Google Calendar.

**Notion**: Popular to-do list app that lets you create lists of tasks to complete and track your progress.

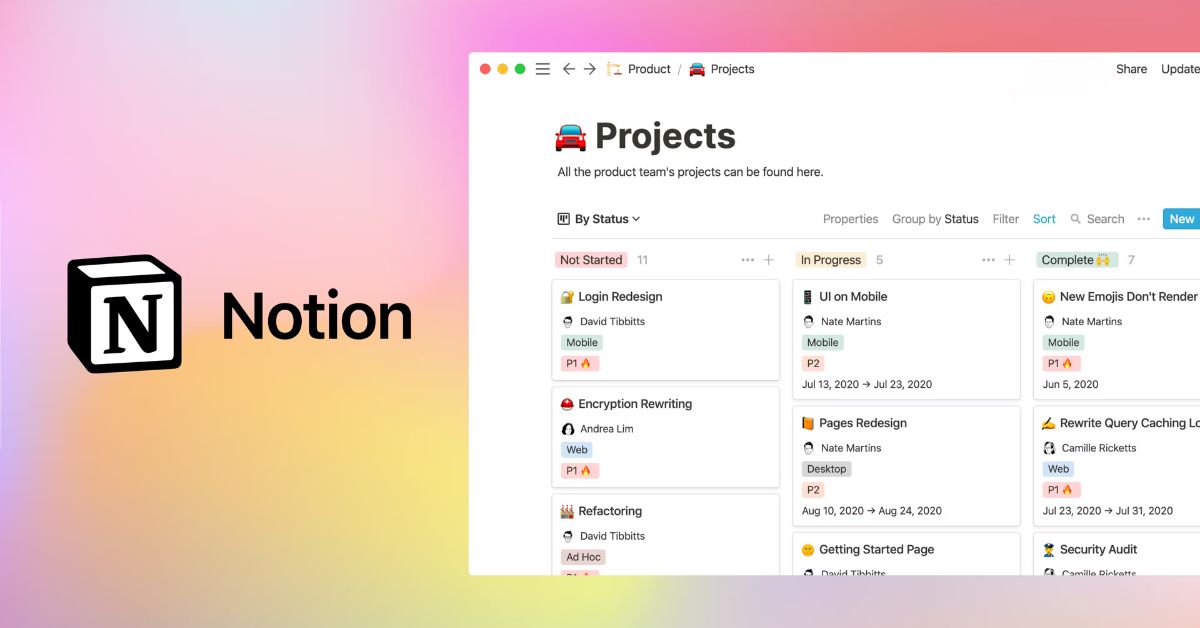


Figure 2.1 4: Notion.

**Trello:** Popular collaboration app that lets you communicate with your colleagues in real time.



Figure 2.1 5: Trello.

**Microsoft teams**: Popular collaboration app that allows you to communicate and collaborate with your colleagues.

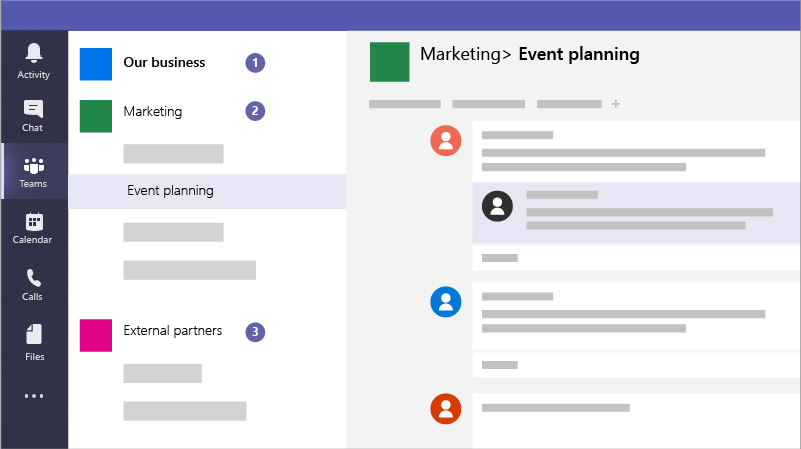


Figure 2.1 6: Microsoft teams.

The choice of construction direction and application operation will depend on the specific needs of the project.

## 2.2 Website design using ReactJS:

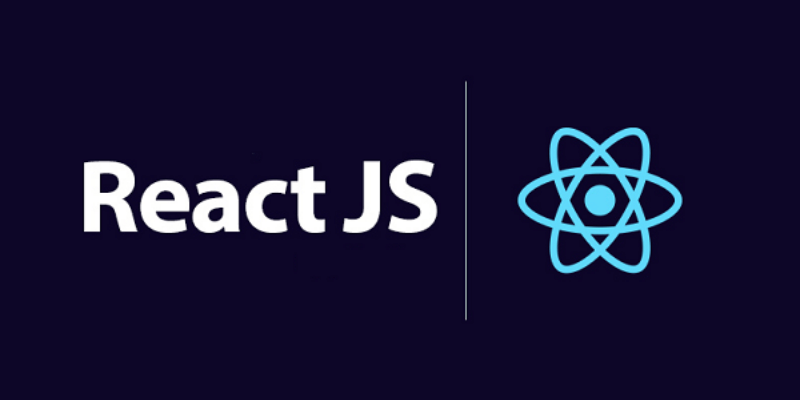


Figure 2.2 1: ReactJS.

**ReactJS** is an open source JavaScript library developed by Facebook in 2013 with the purpose of making it fast, efficient, and easy to build web user interfaces. The unique feature of ReactJS is the use of a virtual DOM model (Virtual DOM) to help optimize rendering performance and minimize errors during development..

Before ReactJS was born, building web interfaces with "vanilla JavaScript" and Jquery faced many limitations such as:

* Difficulty in managing state and updating the interface.
* Rendering performance is slow and resource intensive.
* Difficulty in code reuse and application maintenance.

ReactJS has solved these problems by providing a new programming model that helps:

* Speed up application development: ReactJS uses a virtual DOM model to help render interfaces faster and minimize errors.
* Improved performance: ReactJS only updates the changed interface parts, helping to save resources and increase application performance.
* Easy to maintain: ReactJS uses reusable components, making the code neat and easy to maintain.

**Advantages of ReactJS**

In addition to supporting fast interface construction, limiting errors during coding, and improving website performance, the following advantages make many developers want to use it.:

* Suitable for a variety of website types.
* Component reuse
* Can be used for mobile applications as well
* Debug easily.
* The community grows large and strong.
* There are many supporting libraries and tools.

## 2.3 MongoDB database:



Figure 2.3 1: MongoDB.

Before learning about MongoDB, let's learn a little about the concept of NoSql. NoSql (abbreviated as None-Relational SQL) is developed on Javascript Framework with JSON data type and key and value data format.

With NoSql you can expand data without worrying about things like creating foreign keys, primary keys, and constraint checking.

NoSql bypasses data and transaction integrity in exchange for fast performance and scalability.

MongoDB is an open source database management system, a NoSql database and used by millions of people.

Compared to RDBMS (Relation Database management system), in MongoDB collection corresponds to table, and document corresponds to row.

As a document-oriented database, data is stored in JSON documents in tabular form like a relational database (SQL), so queries are very fast because related information is stored together.

# CHAPTER 3: RECOMMENDED MODEL

## 3.1 Overview of the recommended model:

Build an effective event management website, meeting the event management and organization needs of the IT department.

The design will be based on two popular calendar and event management software: Google Calendar and Notion.

## 3.2 Specific functions:

* Create, edit, delete events.
* Displays the specific time of the event on the calendar.
* Notify event details to participants such as: time, location, ... via previously registered email.
* Statistics on the number of participants, and report when the event ends.

## 3.3 Technology used:

* User interface: ReactJS, Redux Tool, React-Big-Calendar.
* Back end: Nodejs, Node-Mailder, Node-cron, Exceljs, …
* Database: MongoDB.

## 3.4 ERD model:

**a. Use Case Diagram.**

**The system will be divided into 3 roles:**

* Students.
* Teacher.
* Admin.

**Student role in the system:**

* Log in to the system.
* View information about upcoming events.
* Register via the google form link embedded in the event content.
* Receive event notifications.
* View your training points and event participation history.

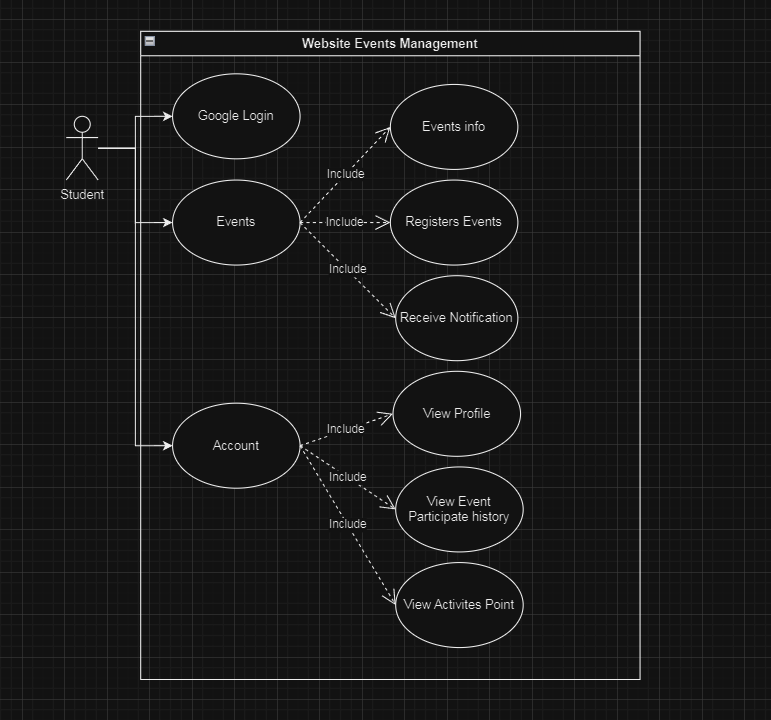


Figure 3.4 1: Relational model of student role.

**Teachers role in the system:**

* Log in to the system
* Create, edit, delete, and view event information.
* Add google form link for students to register.
* Add registered students, students participating in the event.
* Receive event reminders.
* Receive event reports when the event is completed.

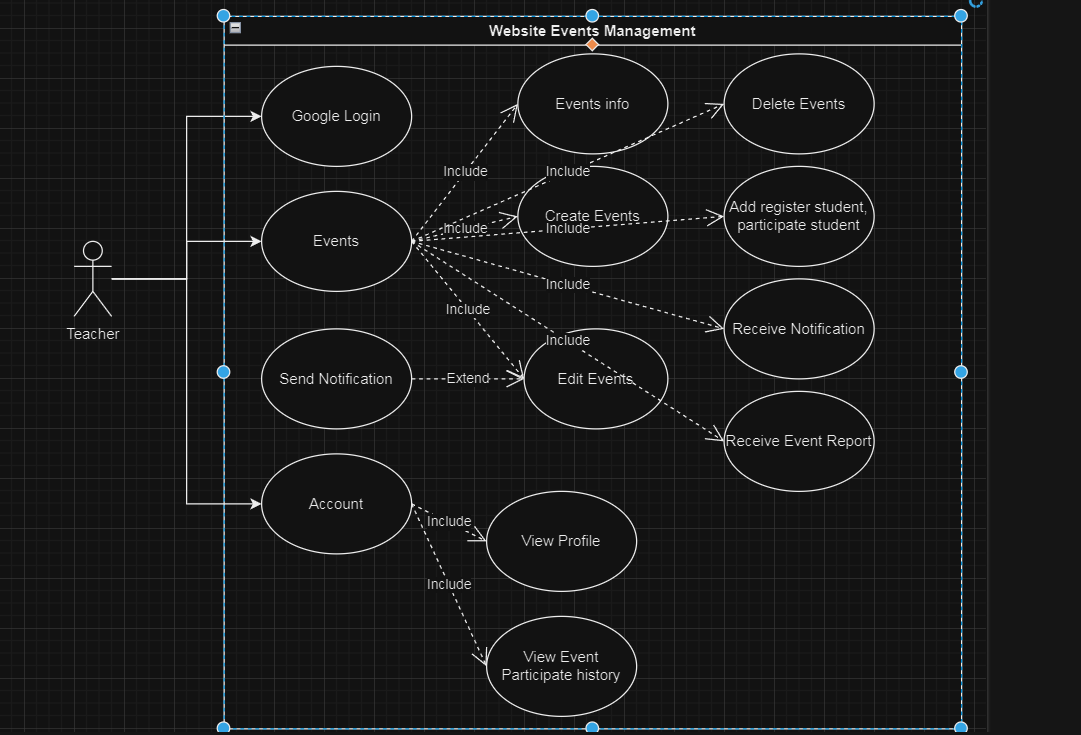


Figure 3.4 2: The relational model of teacher role.

**Admin role in the system:**

* Log in to the system
* Create, edit, delete, and view event information.
* Add google form link for students to register.
* Add, edit, delete, and assign user rights in the system.
* Add registered students, students participating in the event.
* Receive event reminders.
* Receive event reports when the event is completed..



Figure 3.4 3: Relational model of the Admin role.

**b. Class Diagram.**

The diagram shows the relationships of objects in the system.

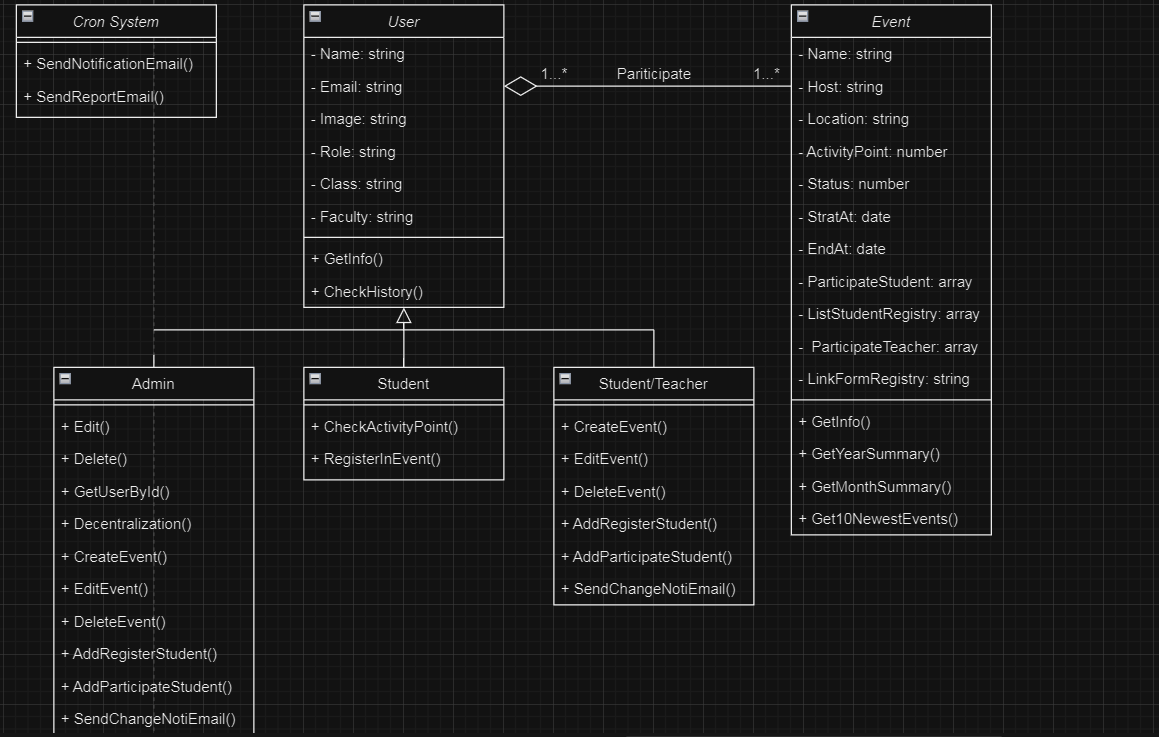


Figure 3.4 4: Class Diagram.

**c. Sequence Diagram.**

The diagram shows the sequential steps to perform functions in the system.

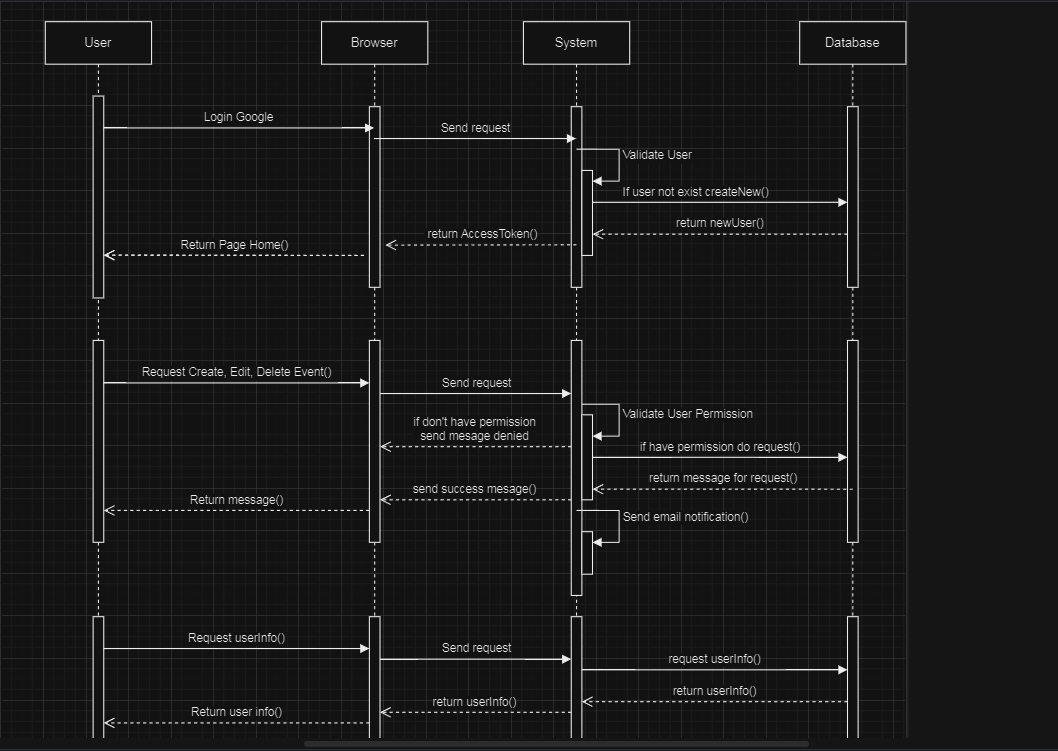


Figure 3.4 5: Sequence Diagram.

**d. Activity Diagram.**

Logic of how the system login process works.

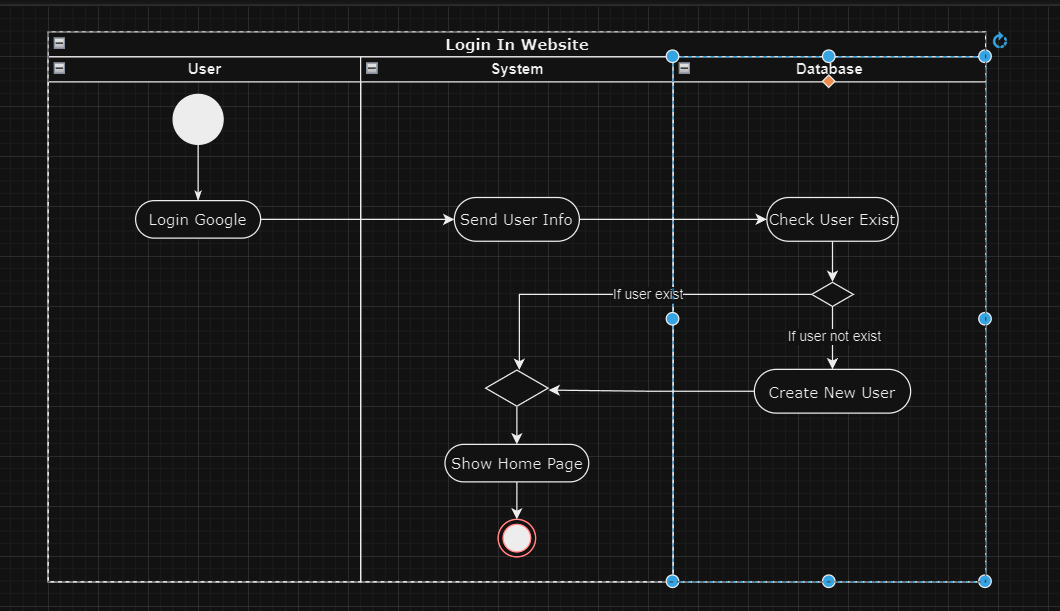


Figure 3.4 6: Login Activity Diagram.

Operational logic views user information and event participation history.

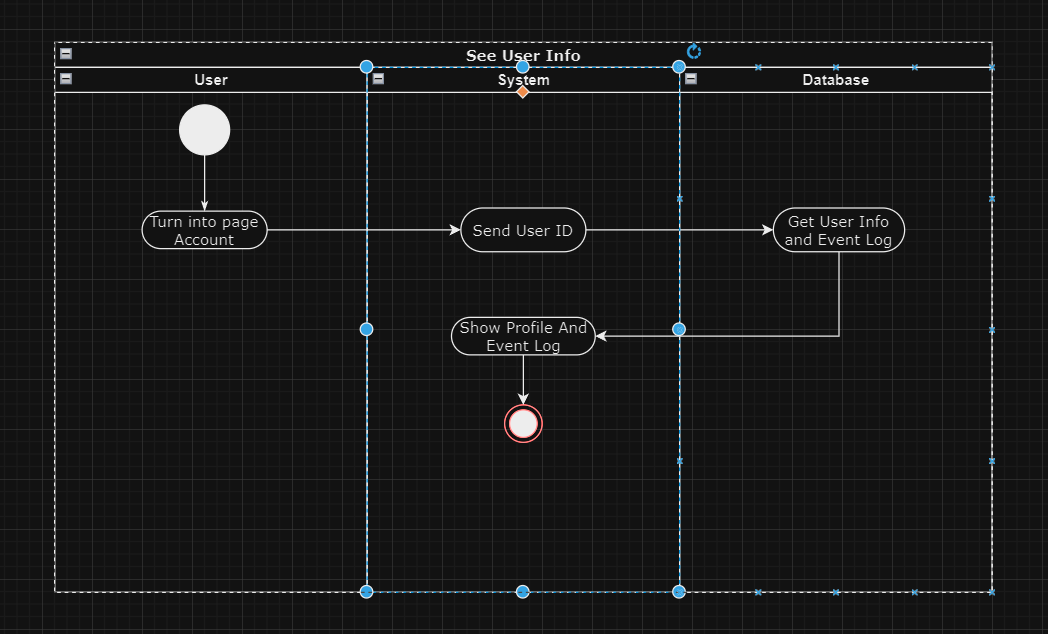


Figure 3.4 7: Get User Info Actvity Diagram.

Logic of how creating, editing, and deleting an event in the system works.

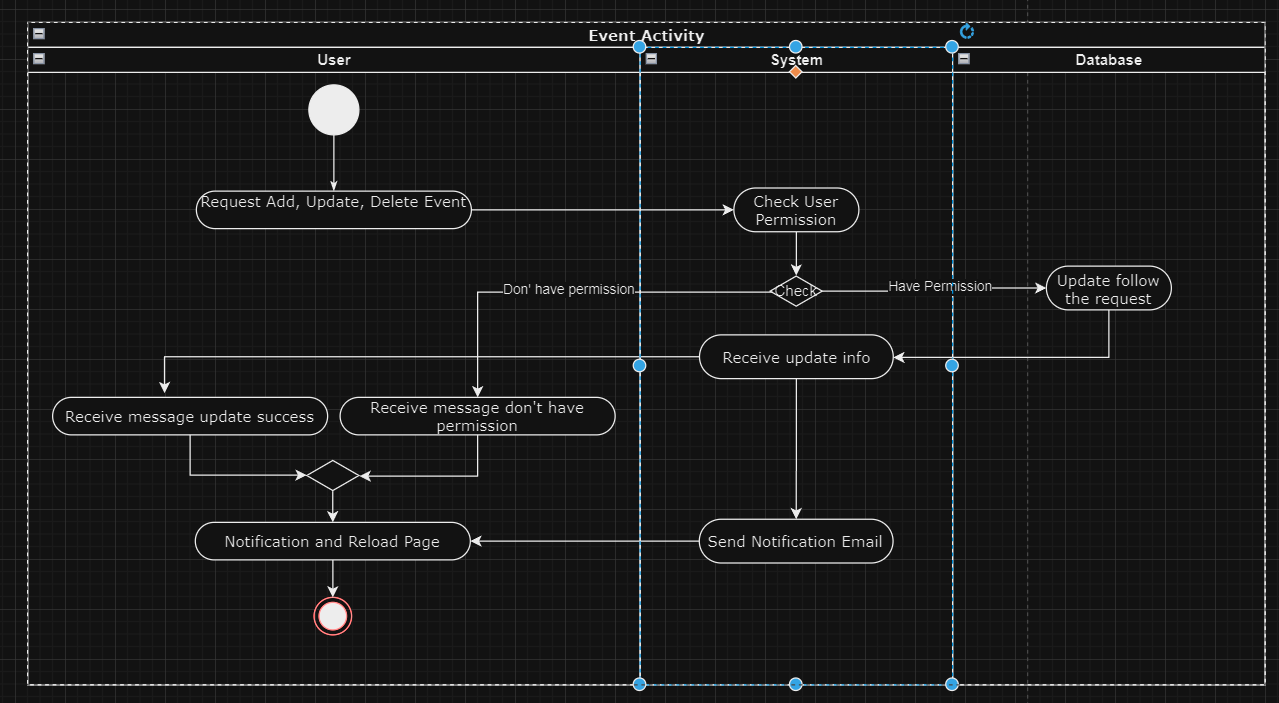


Figure 3.4 8: Request Event Activity Diargram.

Logic of how authorization, updating and deleting users in the system works.

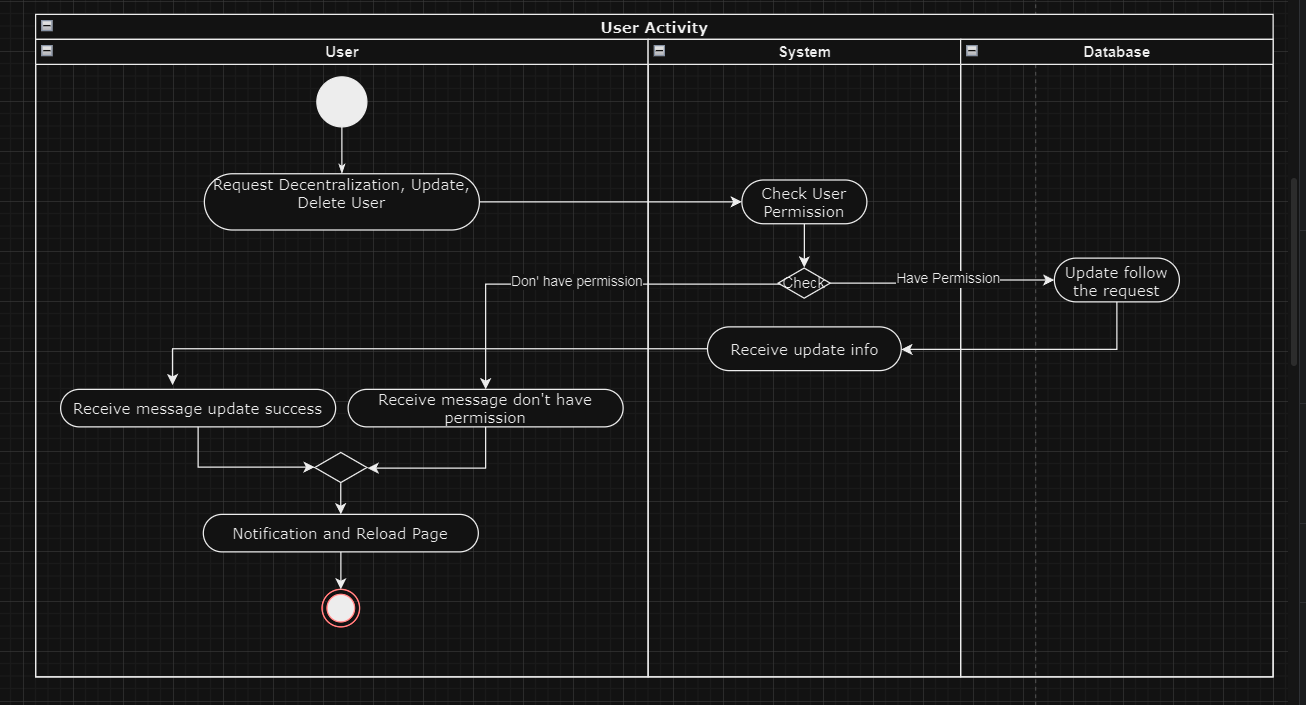


Figure 3.4 9: Request Edit User Activity Diagram.

# CHAPTER 4: WORK SHARING.

## 4.1 Work sharing Backend:

Table 4.1 1: Work sharing Backend.

|  |  |
| --- | --- |
| **Work in the system** | **The person in charge of the job** |
| Build the system's ERD. | Pham Vu Quoc Cuong. |
| Implement the code of the google login section. | Tran Pham Anh Tuan. |
| Implement cron node code to automatically notify. | Pham Vu Quoc Cuong. |
| Implement the code of the user. | Tran Pham Anh Tuan. |
| Implement the code of the event. | Pham Vu Quoc Cuong. |
| Do import, export student registration and participation data using excel. | Tran Pham Anh Tuan. |
| Send email notifications to all participants when events are created, edited, etc. | Pham Vu Quoc Cuong. |

## 4.2 Work sharing Fontend:

Table 4.2 1: Work sharing Fontend.

|  |  |
| --- | --- |
| **Work in the system** | **The person in charge of the job** |
| Login page with google. | Tran Pham Anh Tuan |
| Home page calendar tab. | Pham Vu Quoc Cuong |
| Home page work process tab. | Tran Pham Anh Tuan |
| Dashboard page summarizes information. | Pham Vu Quoc Cuong |
| User management page. | Tran Pham Anh Tuan |
| Event management page. | Pham Vu Quoc Cuong  Tran Pham Anh Tuan |
| User information page profile tab | Tran Pham Anh Tuan |
| User information page  event history tab. | Pham Vu Quoc Cuong |

## 4.3 Work sharing report:

Table 4.3 1: Work sharing Report.

|  |  |
| --- | --- |
| **Reporting work** | **The person in charge of the job** |
| Word Report | Pham Vu Quoc Cuong |
| Powerpoint Report | Tran Pham Anh Tuan |

# CHAPTER 5. EXPERIMENTATION

## 5.1 Login Page:

**a. Description:**

This page allows students and teachers to access the website that manages the system’s events. Users will log in with their Google Account; if it’s their first time logging in, the backend will retrieve data from the Google account to create a new user account with the default role of a student.

**b. Interface**:

If users have not logged into the system and attempt to access other pages, they will be redirected to this page until they successfully log in. The system will store the user’s access token in a Cookie, which will expire after 7 days.

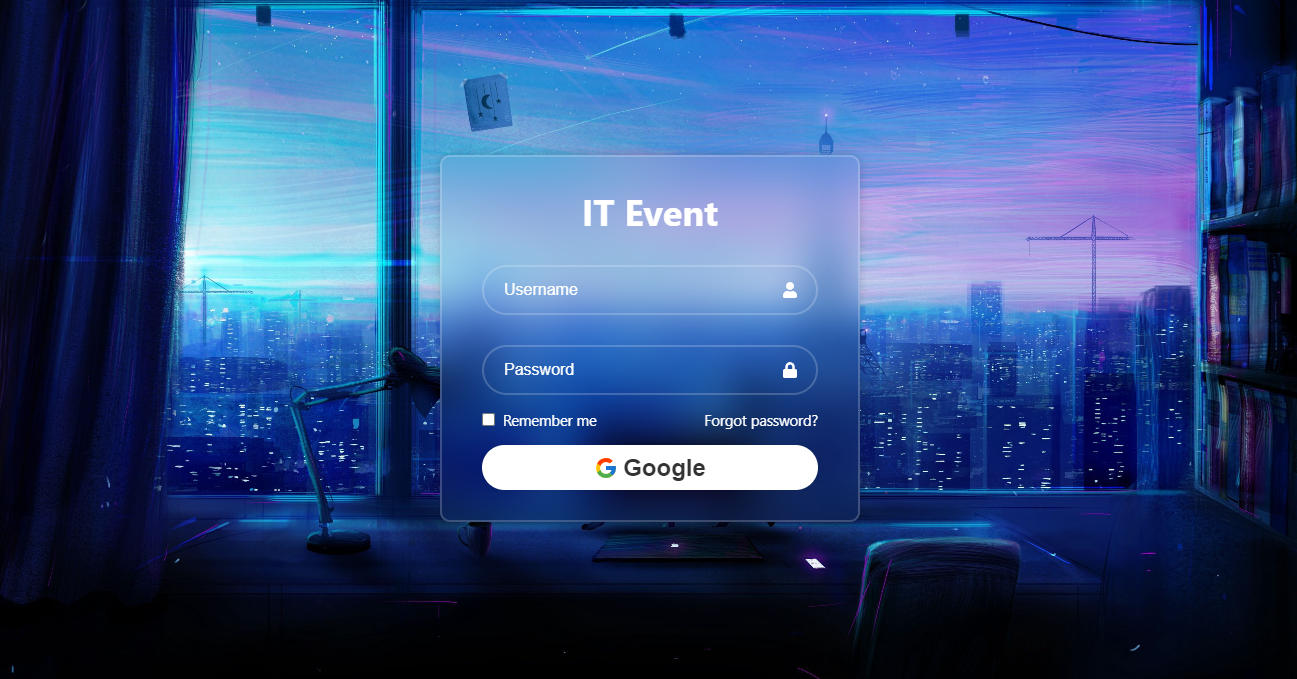


Figure 5.1 1: UI page login.

When clicking on the Google button, it will call the google api so the user can choose their account.

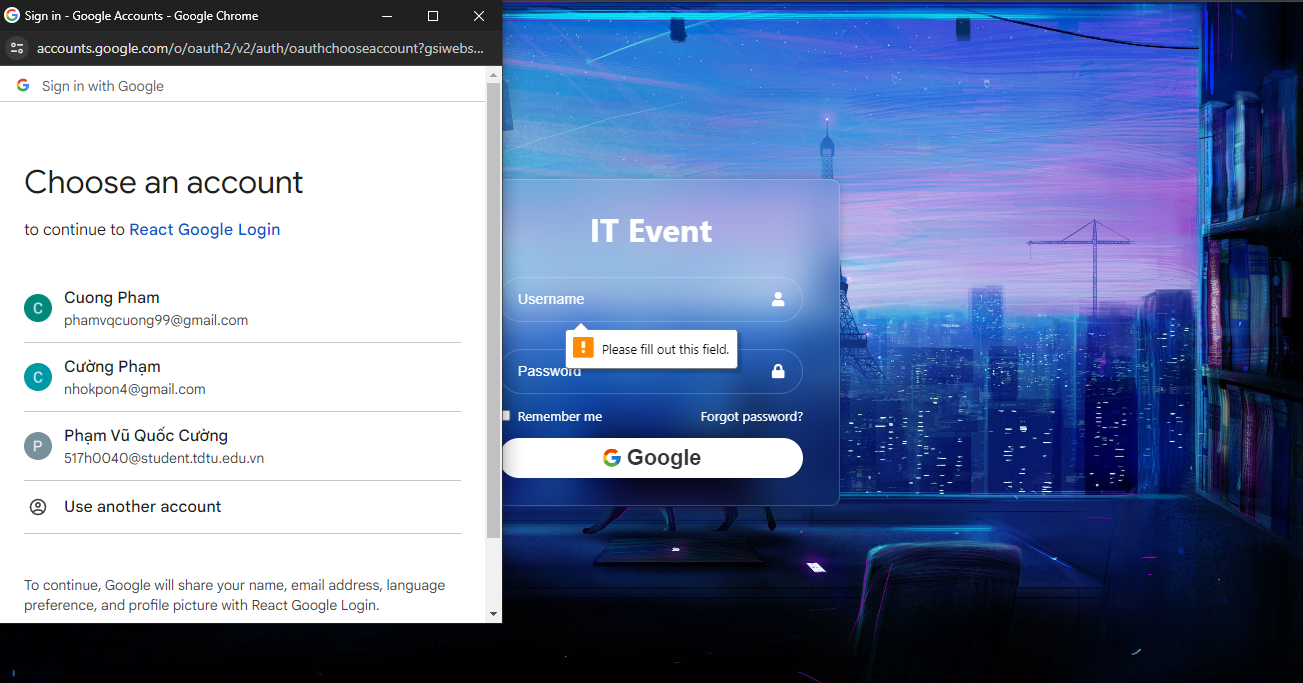


Figure 5.1 2: Sign in with google account.

**c. Operating code:**

After the user logs in with Google, their information is sent to the backend to determine if this user already exists and to create an access token. This process typically involves checking the user’s unique Google ID against the database and, if the user is new, creating a new user record with the provided information. The access token is then generated, which will be used to authenticate subsequent requests by the user within the application.

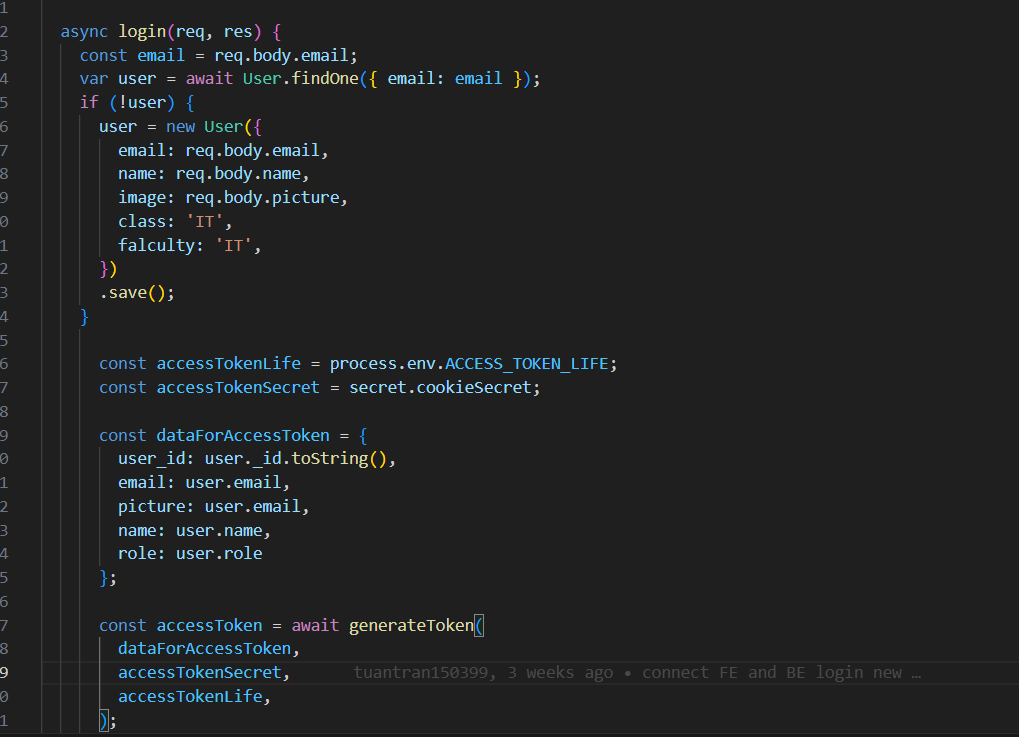


Figure 5.1 3: Backend code validates user 1.

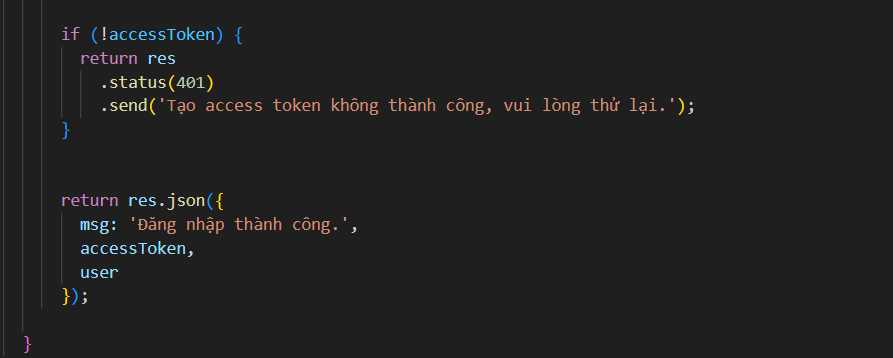


Figure 5.1 4: Backend code validates user 2.

## 5.2 Account page:

**a. Description:**

This is a page that allows users to see their specific information. If there are any errors, they can notify the admin so they can edit the information.

**b. Interface:**

The first tab will show the user profile information such as name, email, class, department, and role of the user in the system.

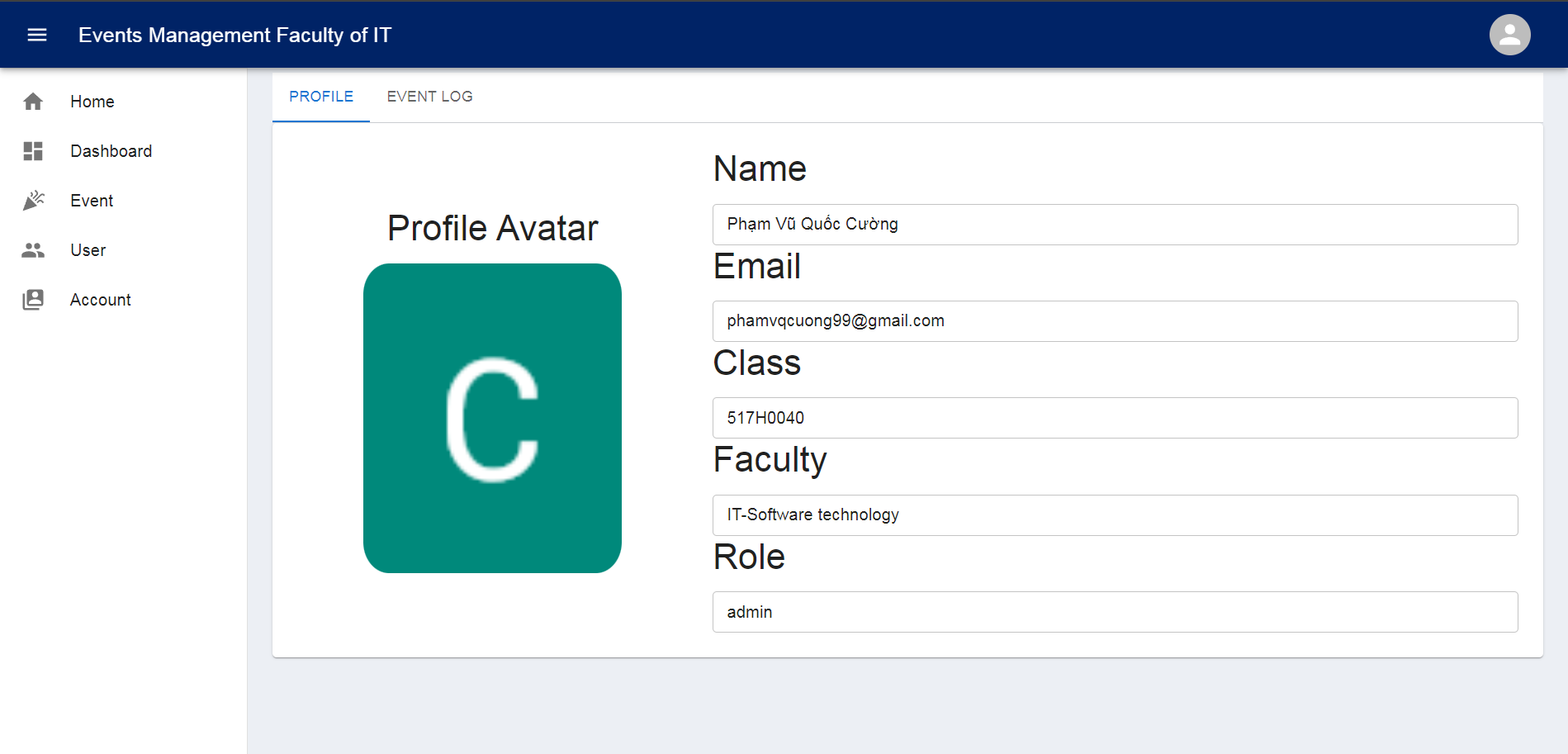


Figure 5.2 1: User profile tab.

The second tab will show the user the events that the user has participated in. Total number of events participated in during the year and total training points earned during the year.

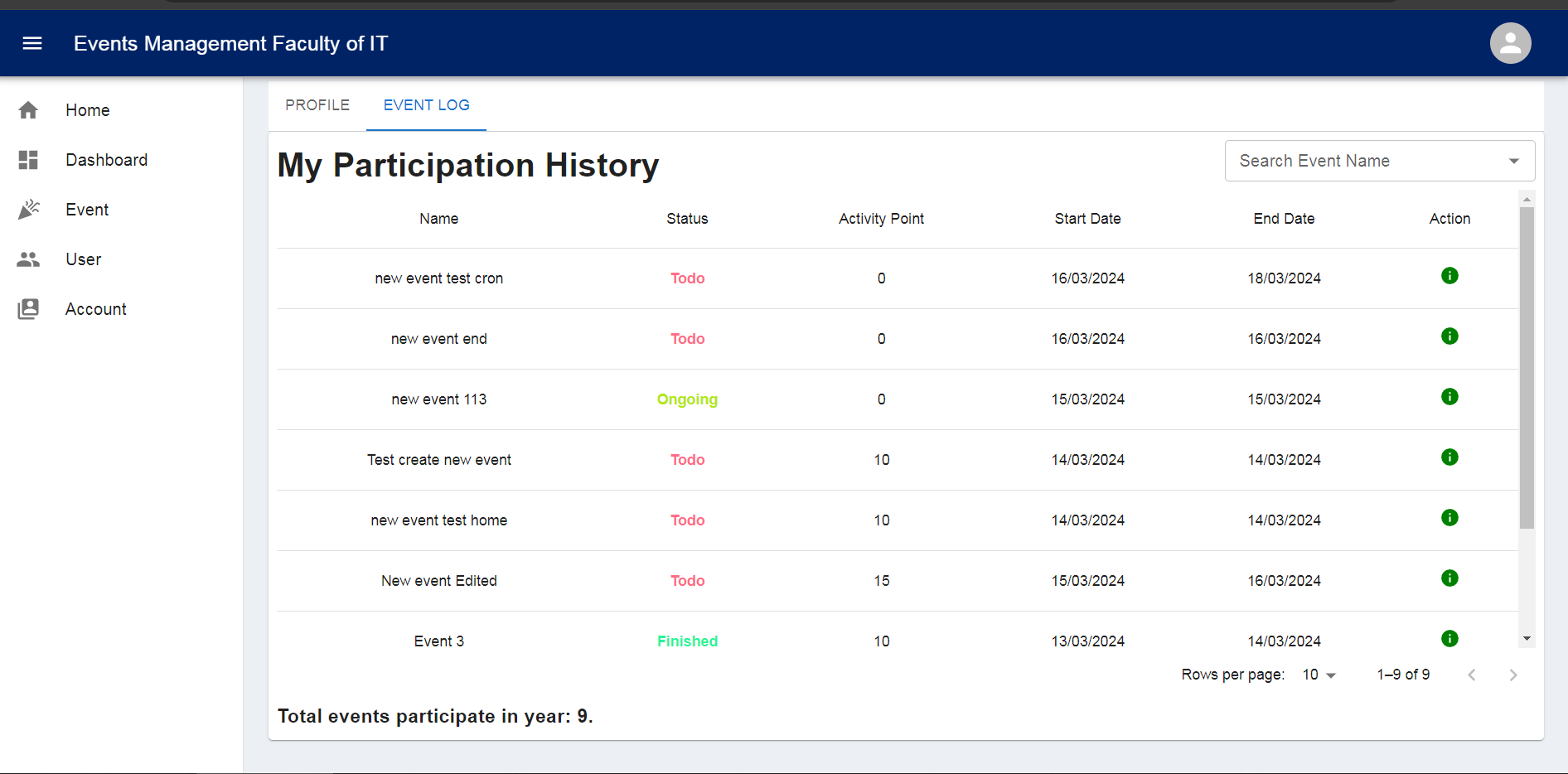


Figure 5.2 2: Join event history tab.

**c.** **Operating code:**

The system will get the user\_id from the access token and send it to the Backend to get the user’s data.

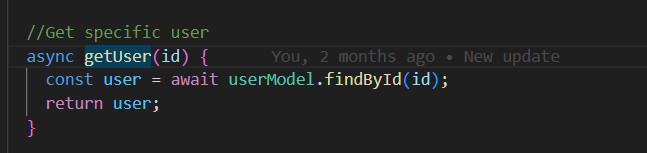


Figure 5.2 3: The backend code retrieves user data.

The system will retrieve all events that the user has ever participated in.

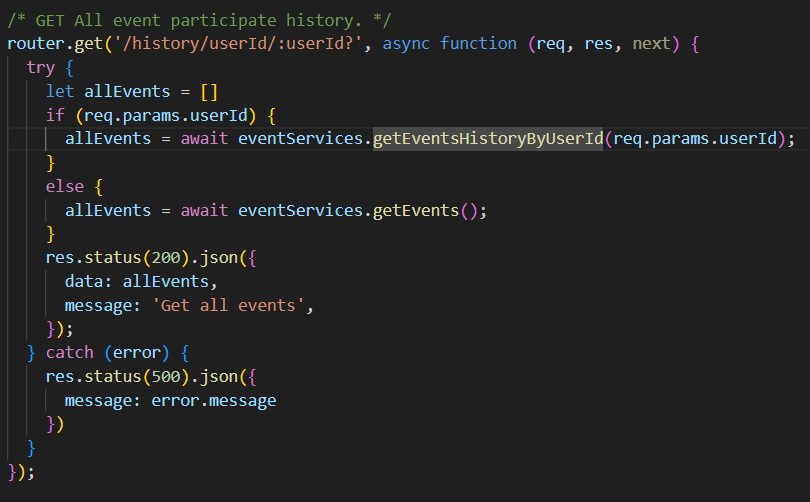


Figure 5.2 4: The backend code retrieves event history data.

The system will retrieve data to get the total events that the user participated in along with the training points that the user achieved during the year.

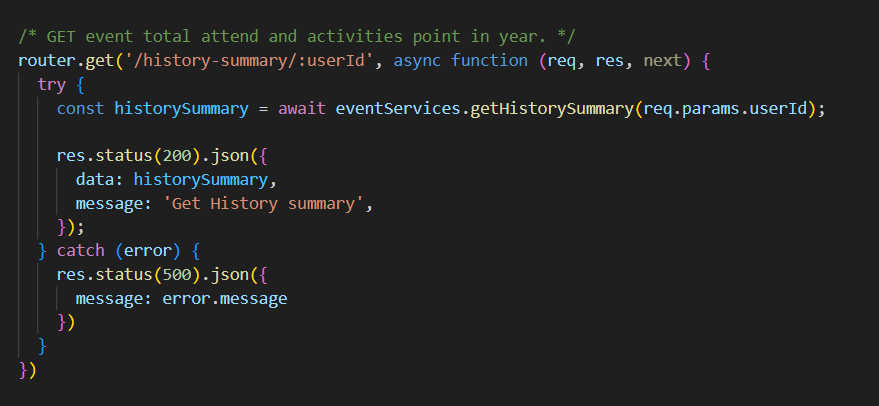


Figure 5.2 5: The backend code retrieves the number of events participated in the year and training points.

**Functionality:**

* This route retrieves historical event data for a user identified by the userId path parameter.
* It interacts with a service eventServices which has a function getHistorySummary to fetch the requested information.

**Purpose:**

* This route provides users with a way to see their past event activity in a summarized format, possibly including:
* Total number of events attended within the year.
* Accumulated points earned from event activities.

## 5.3 Home Page:

**a. Description**

This is the page that shows users events for the month, week, and day. Users can create new events, as well as click to edit existing events.

**b. Interface:**

The home page interface includes 2 tabs: 1 tab shows the event on the calendar, 1 tab shows the completion level of the event.

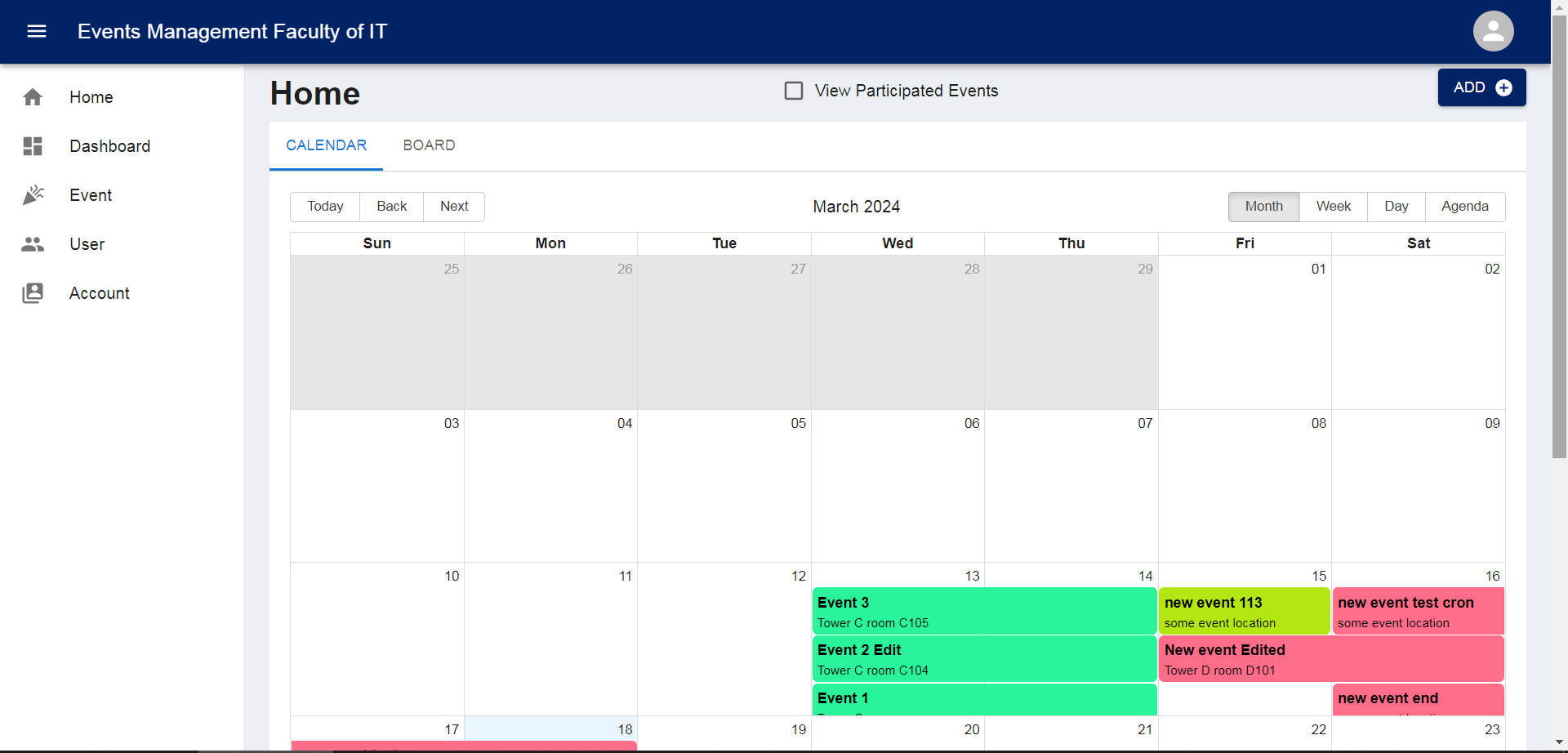


Figure 5.3 1: Tab event on calendar.

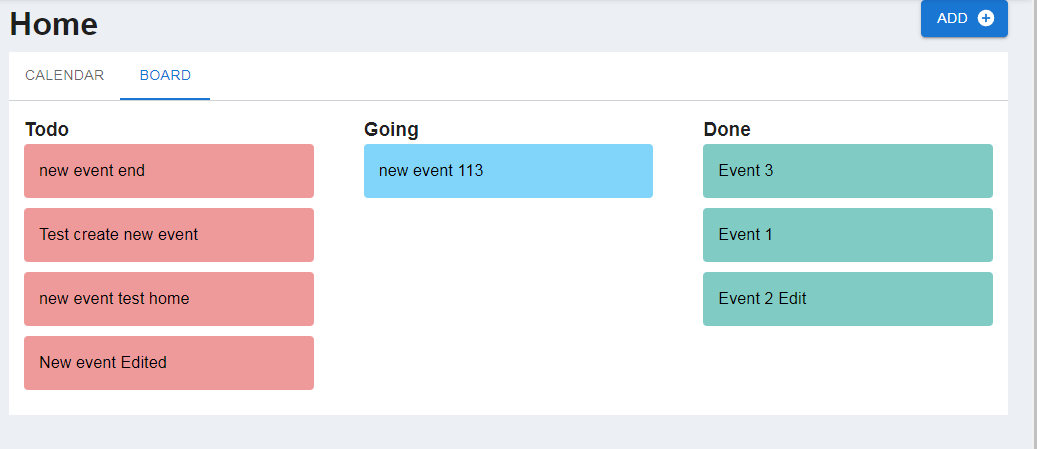


Figure 5.3 2: Event completeness tab

**c. Operating Code:**

FE will call up and get all the events to put into the calendar as well as the work completion sheet.

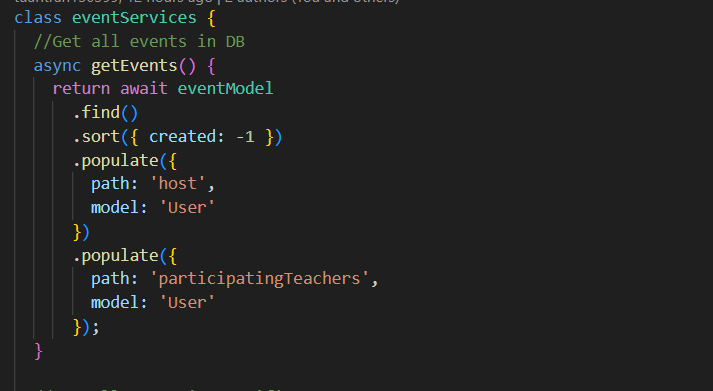


Figure 5.3 3: Get all events in DB.

Users can also filter and view only events they registered for or in which they participated.

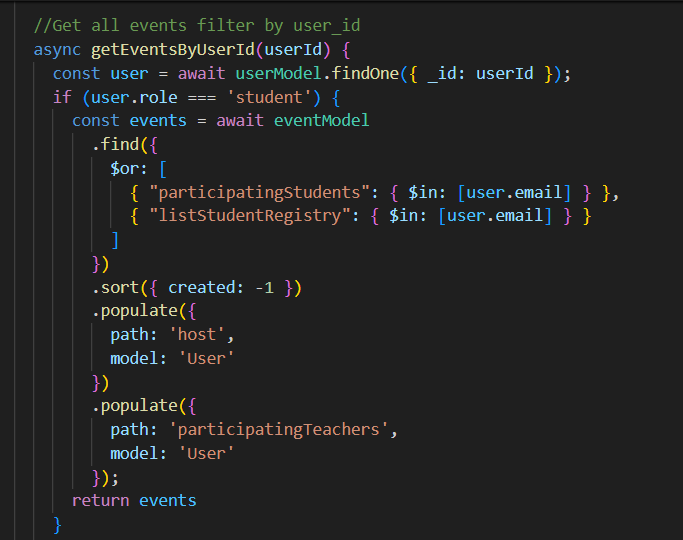


Figure 5.3 4: The code filters events by userId if the role is student.

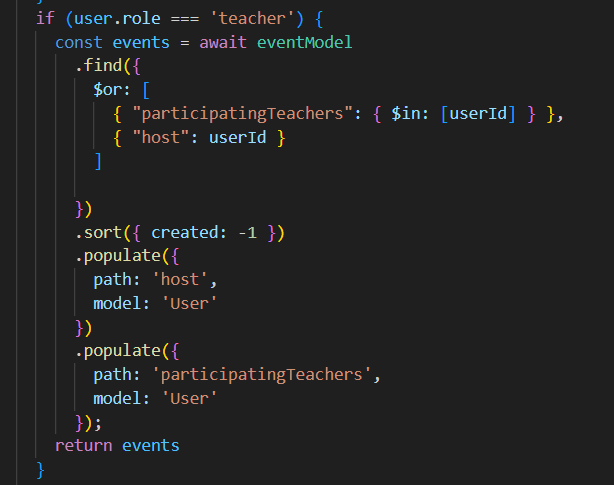


Figure 5.3 5: The code filters events by userId if the role is teacher.

**Explain function technical:**

This function finds events for a user (by ID) and filters them based on role.

Students: Finds events where their email is in either participatingStudents or listStudentRegistry (event registrations).

Teachers: Finds events where they are listed as a participating teacher or the event host.

There’s a small feature logical for admin: if the user is neither a student nor a teacher, the function defaults to returning all events due to the structure of the if...else statements.

## 5.4 Dashboard Page:

**a. Description:**

This is a page that summarizes event information for teachers and admins. About data overview of events in the year, month and 10 latest events created.

**b. Interface:**

The dashboard interface consists of 3 parts:

* Cards show an overview of completed progress and number of events during the year.
* The pie chart shows an overview of progress and number of events during the month.
* The data table shows the 10 most recent events.

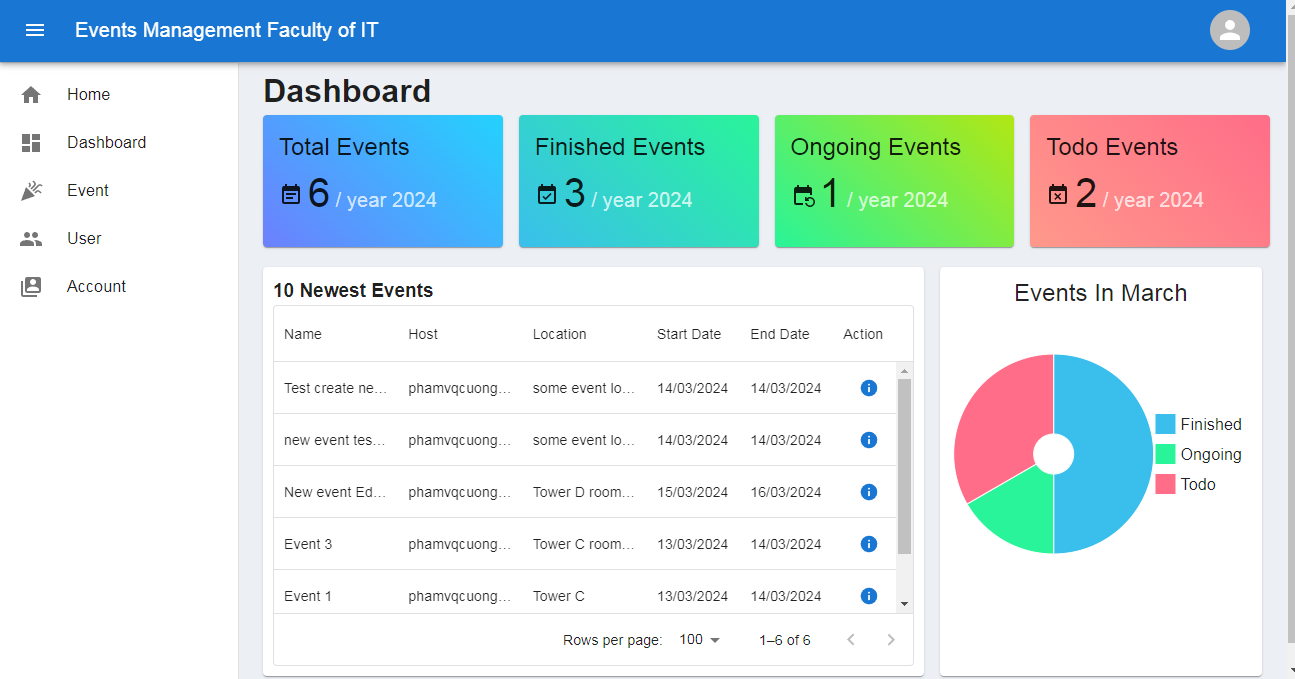


Figure 5.4 1: Interface page of Dashboard.

**c. Code operation**

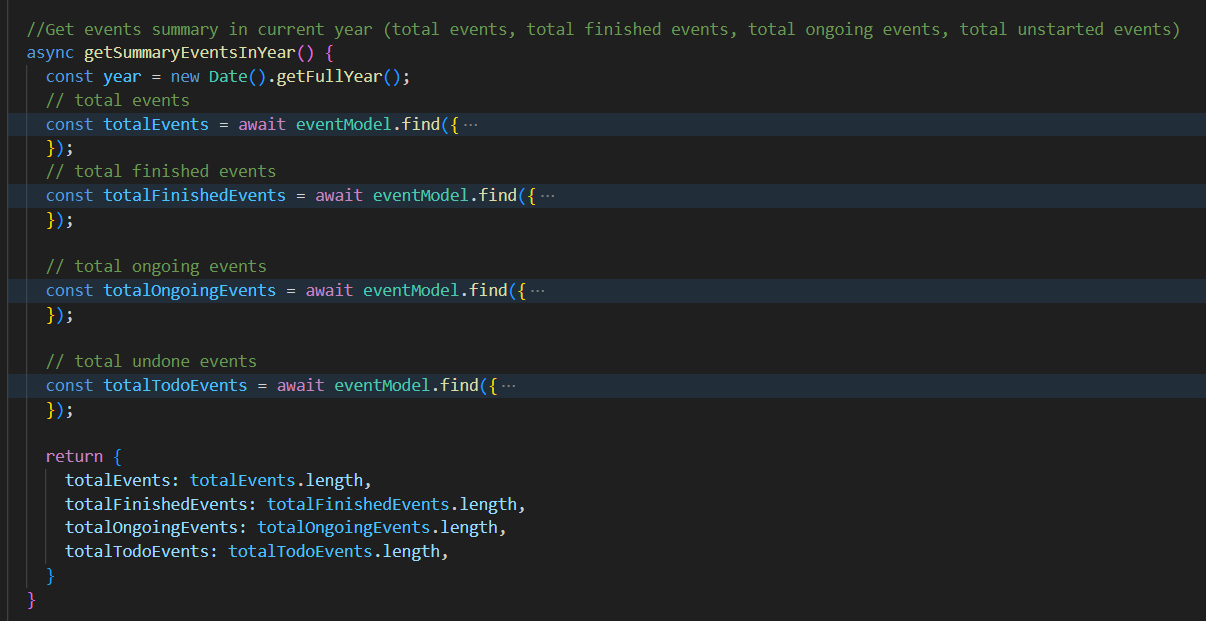
To get data for the dashboard page, you also need 3 APIs to get general data:

Figure 5.4 2: Code to get summary of event in year

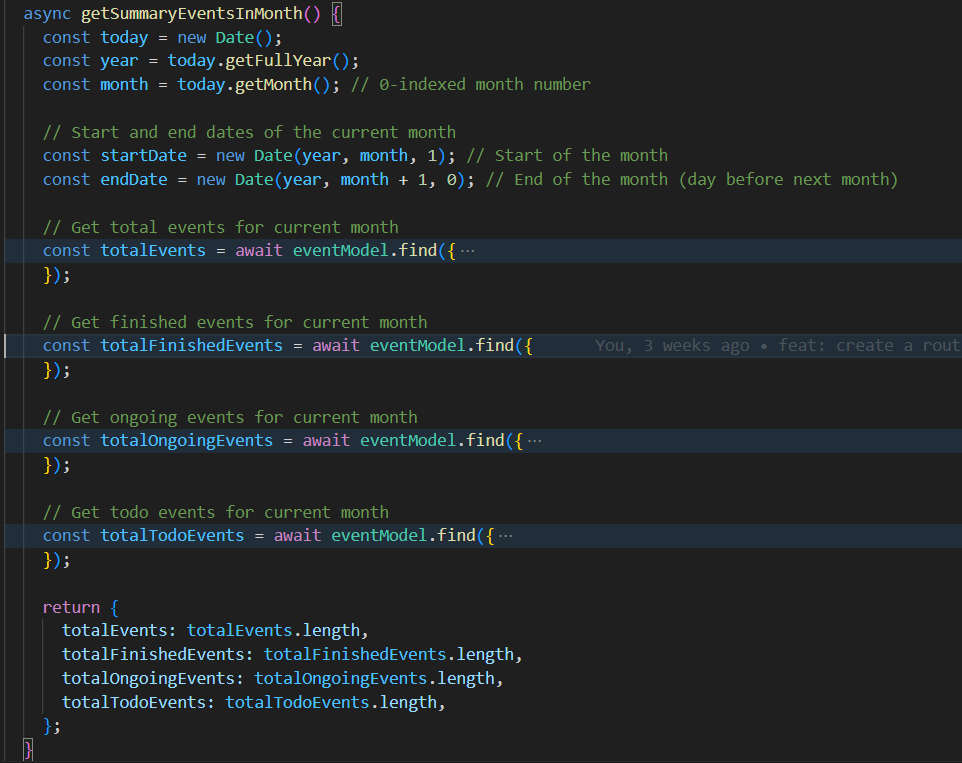


Figure 5.4 3: Code to get summary of event in month

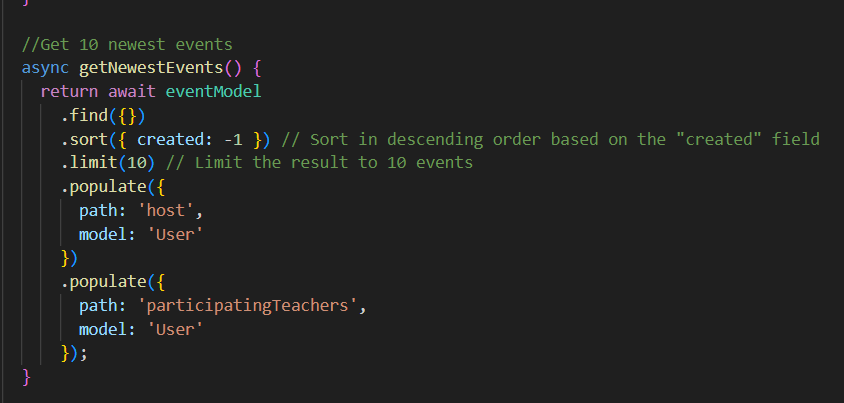


Figure 5.4 4: Code to get 10 newest events.

* **getSummaryEventsInYear():**

Retrieves a summary of events within the current year, categorized by their status (totalEvents, totalFinishedEvents, totalOngoingEvents, totalTodoEvents).

Uses the JavaScript Date object to calculate the start and end of the current year for filtering.

* **getSummaryEventsInMonth():**

Similar to getSummaryEventsInYear, but focuses on the current month.

Calculates the start and end dates of the current month for filtering.

* **getNewestEvents():**

Retrieves the latest 10 events, sorted by the created field in descending order.

Uses the .limit() method to restrict the number of events returned.

Enriches the returned event data with related user information for host and participatingTeachers using .populate().

## 5.5 User Page:

1. **Description:**

This is the user management page. Here the admin can delete and edit user information in the system, as well as assign permissions.

**b. Interface:**

The user management page includes functions such as searching, filtering data by role, editing userinformation and deleting users from the system.

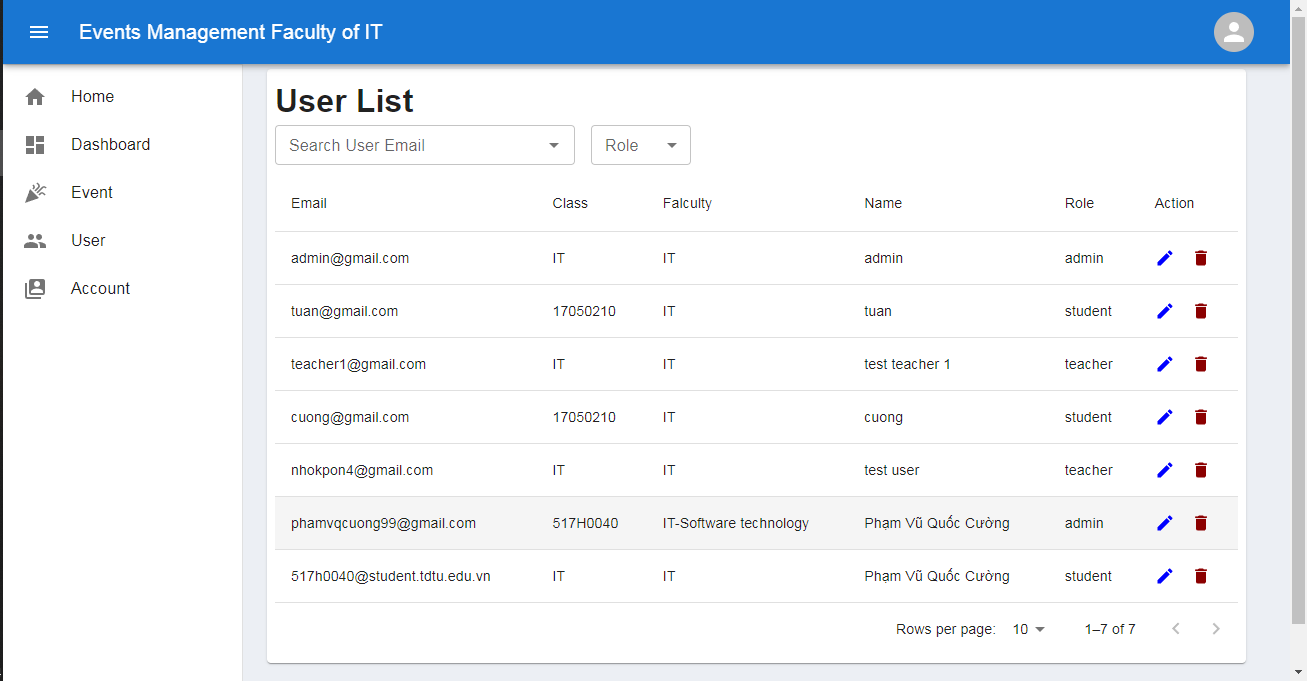


Figure 5.5 1: Interface of user page

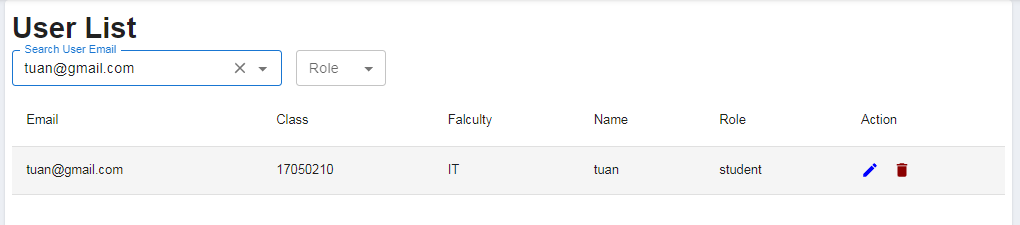


Figure 5.5 2: Searching user infomartion.

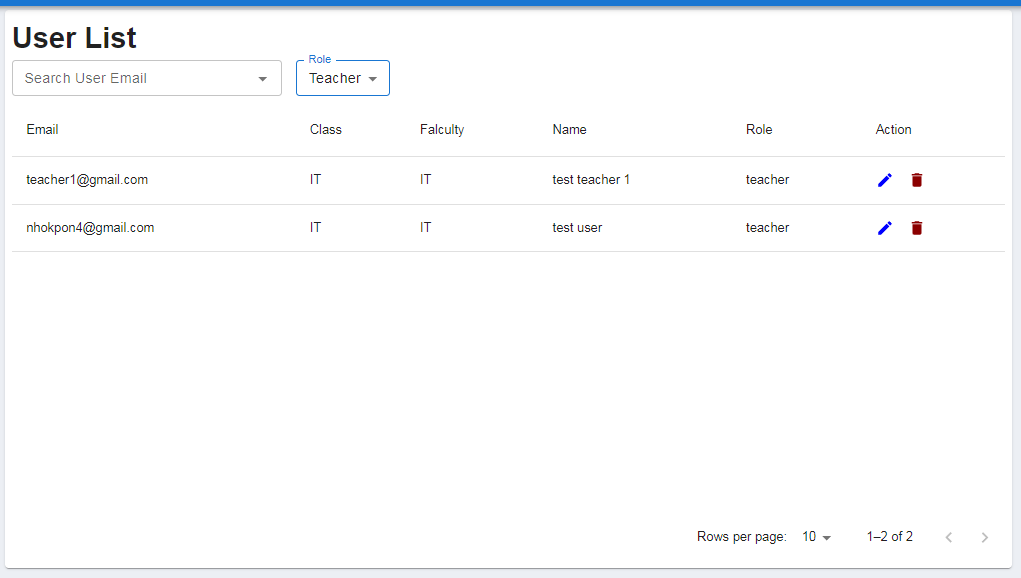


Figure 5.5 3: Filter user.



Figure 5.5 4: Edit user.

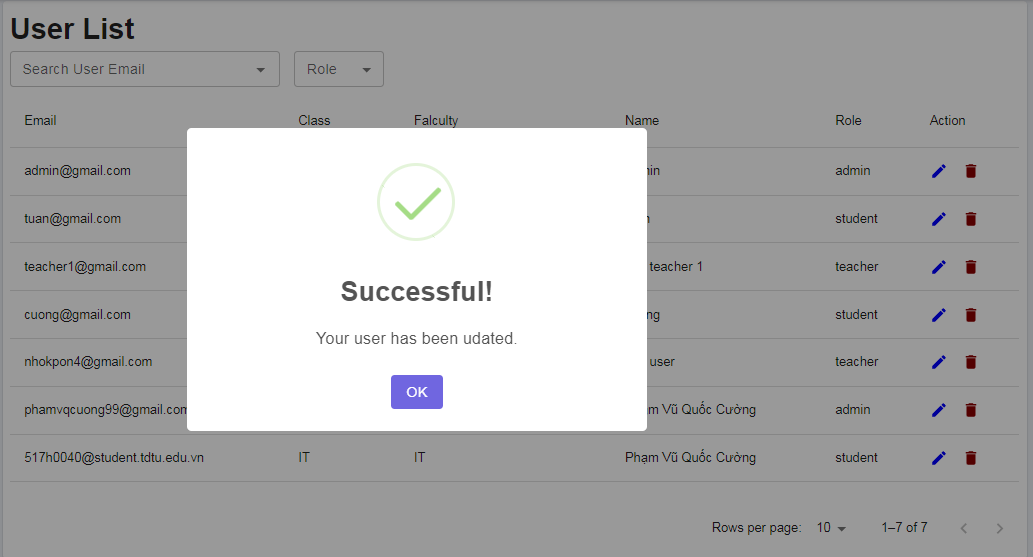


Figure 5.5 5: Notification when user is successfully edited.

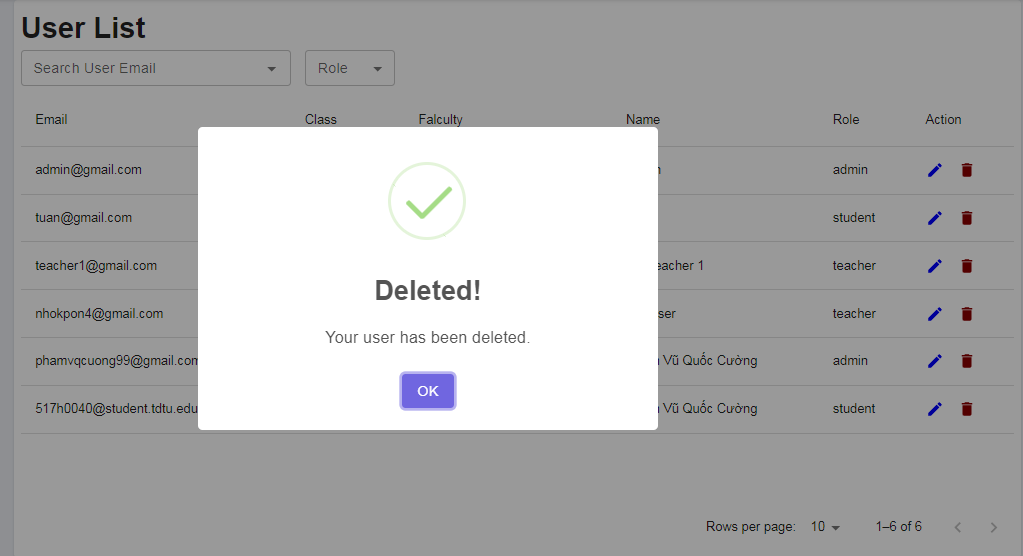


Figure 5.5 6: Notification when user is successfully deleted.

**c. Operating code:**

APIs to create, edit, view information and delete users.

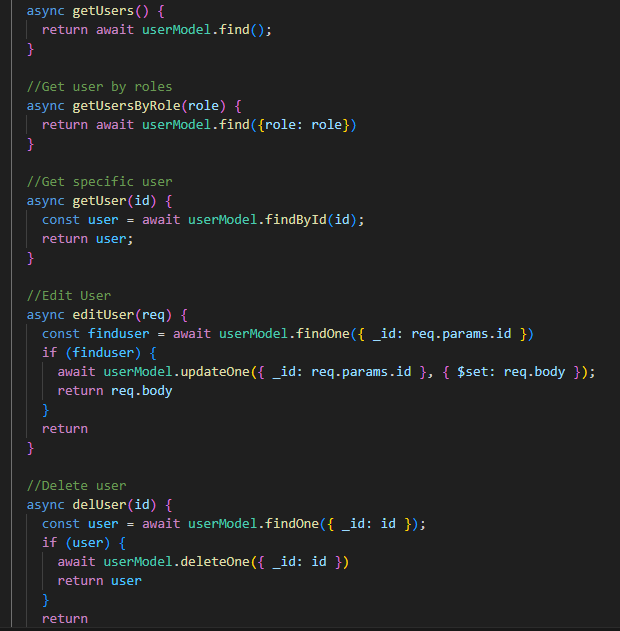


Figure 5.5 7: Code CRUD of user.

## 5.6 Event page:

**a. Description:**

This page serves as the central management hub for all events within the Information Technology Department. Individuals in roles such as professors or administrators have the capability to view information, create new entries, edit, delete, and also search for event names.

**b. Interface:**

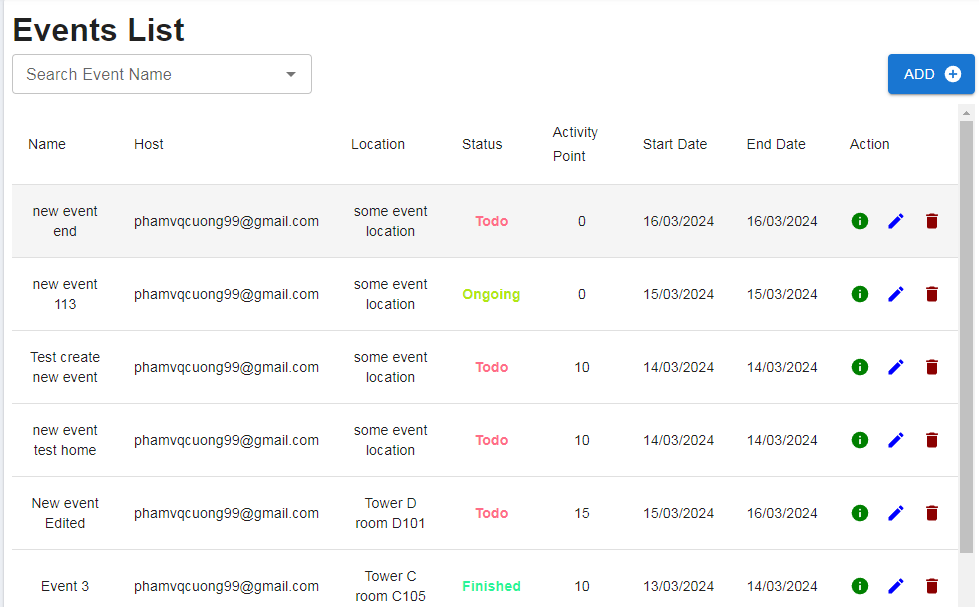
The main interface of the event management page within the system.

Figure 5.6 1: Main interface of event management page.

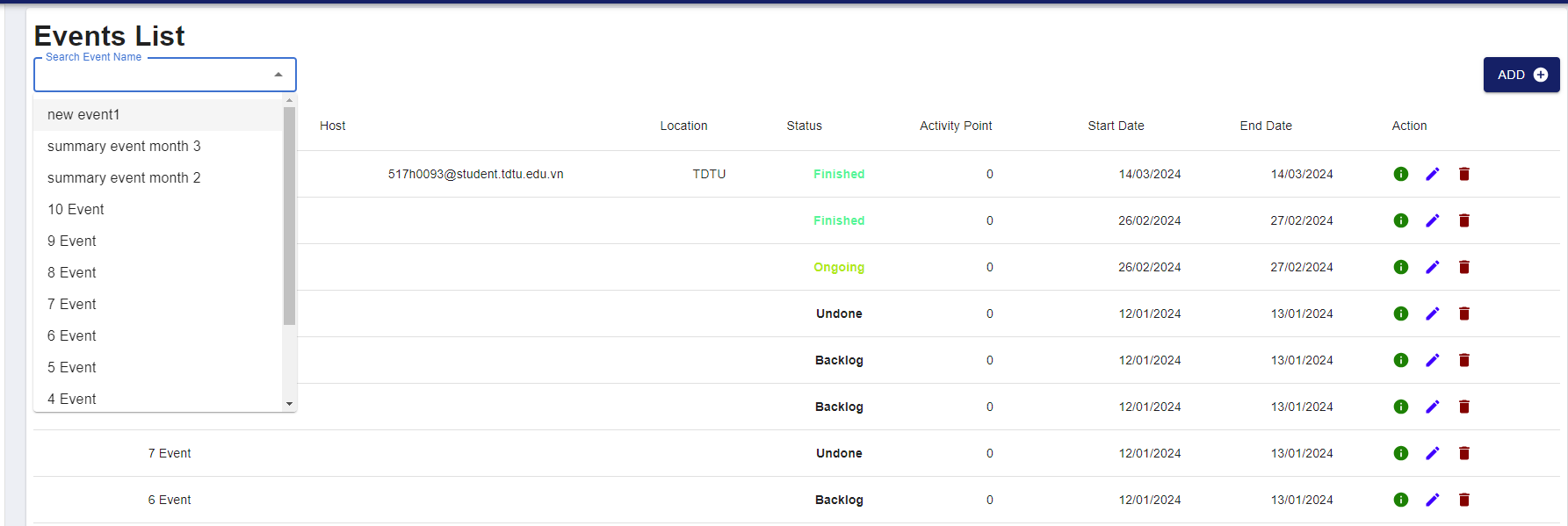
Event search function in UI Event List. 

Figure 5.6 2: Event search function.

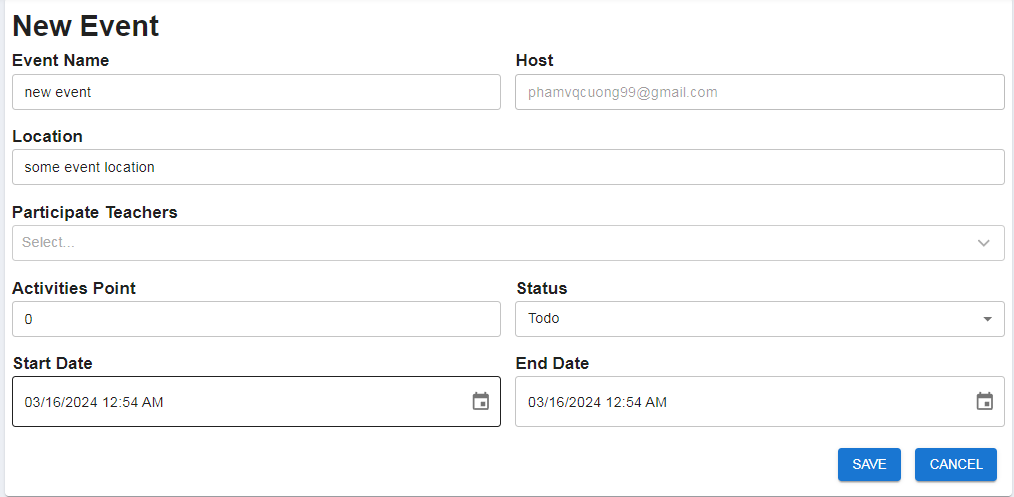
Function to add new events after user click into “add” button.

Figure 5.6 3: Function to add new events.

Send notification emails to event participants.

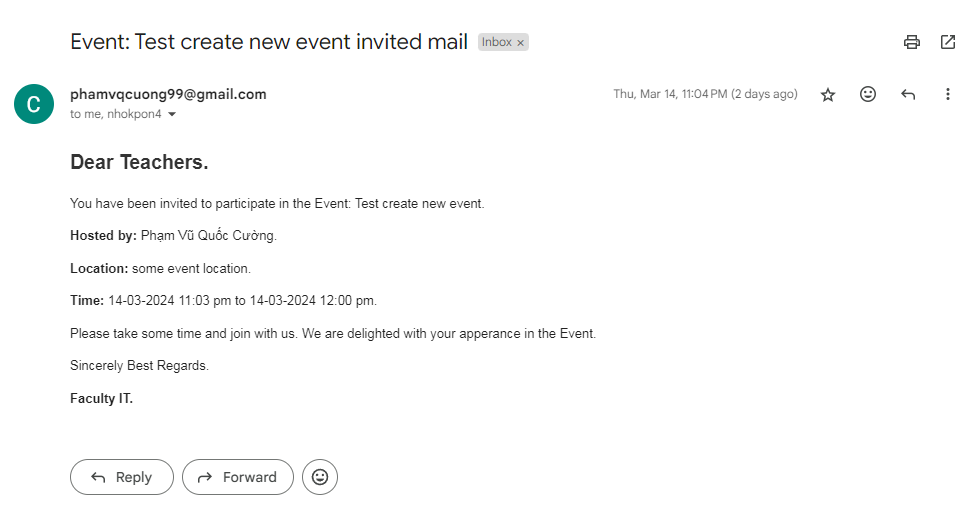


Figure 5.6 4: Email notification of event participation.

View event information, and edit event information. On this page, users can add registered students and students who have participated in the event.

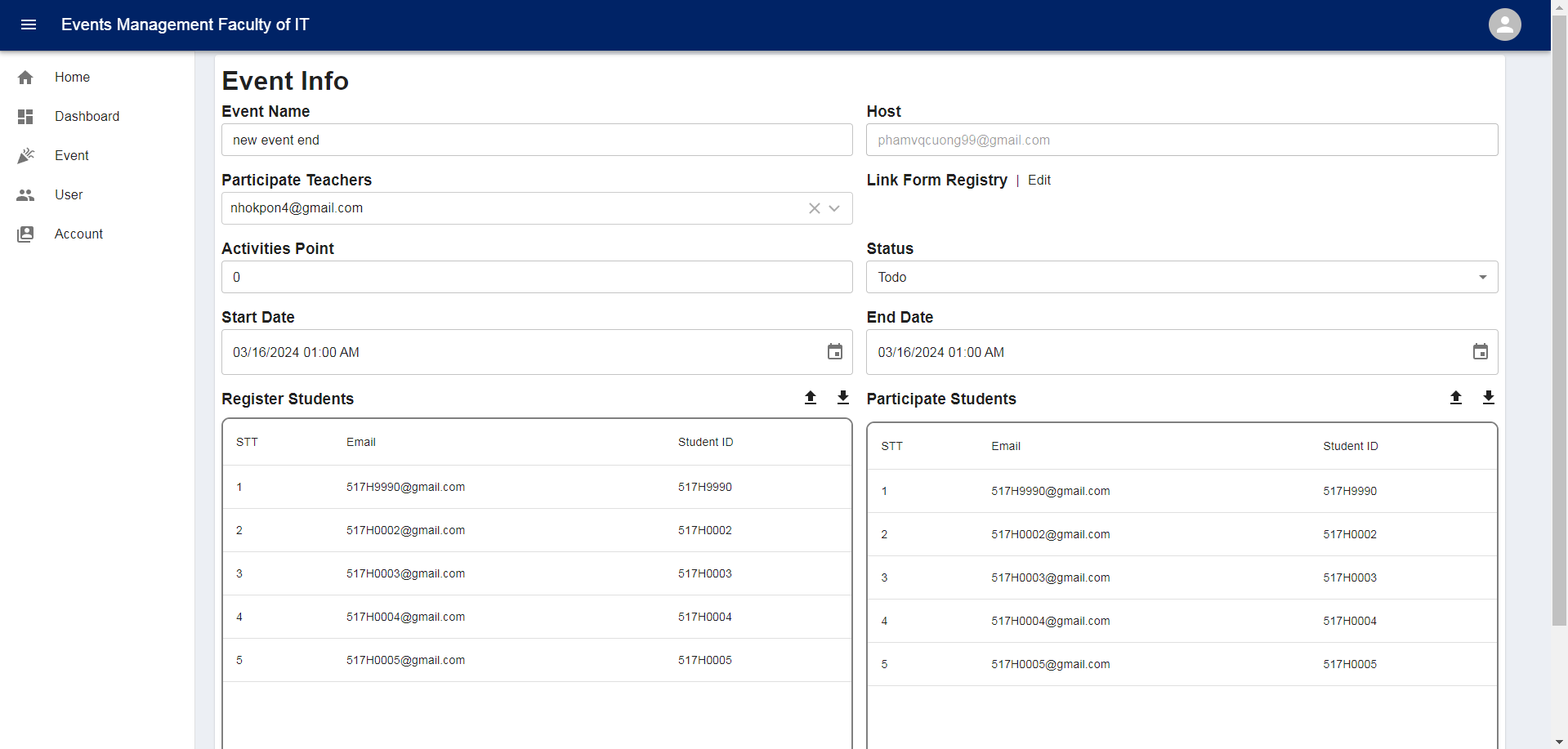


Figure 5.6 5: View and edit event information.

After updating the data and proceeding with the save operation, a notification will appear, prompting the user to choose whether to send an email alert regarding the event changes to the participants.

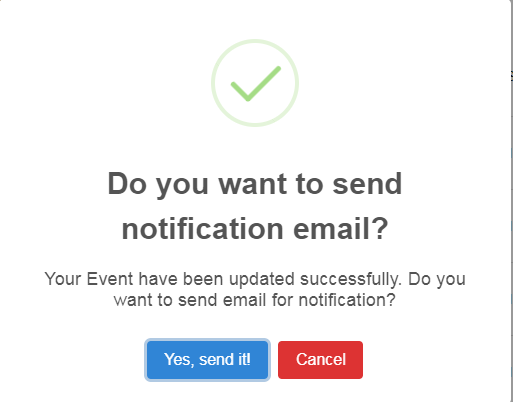


Figure 5.6 6: Email notification when editing event.

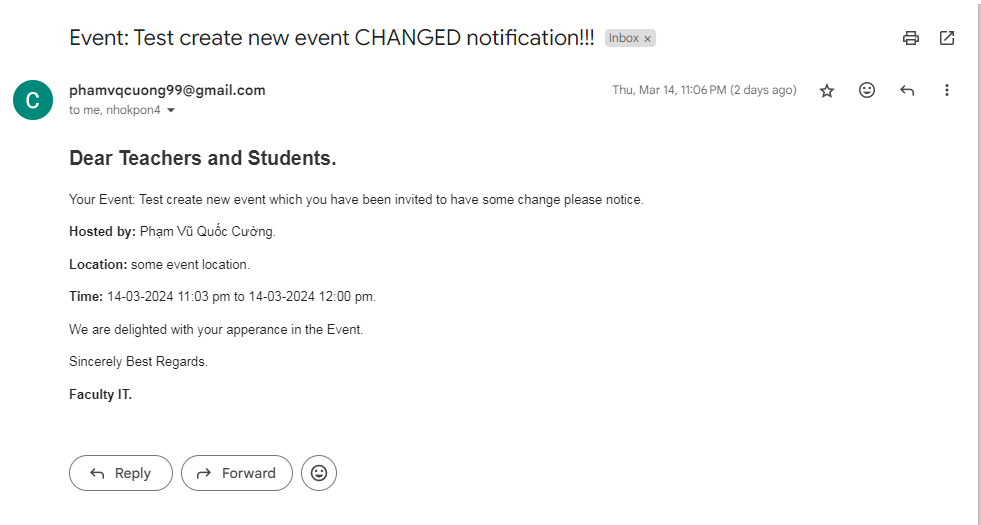


Figure 5.6 7: Email when the user chooses to send a change email.

Users can delete events via the delete button next to the “action” column.

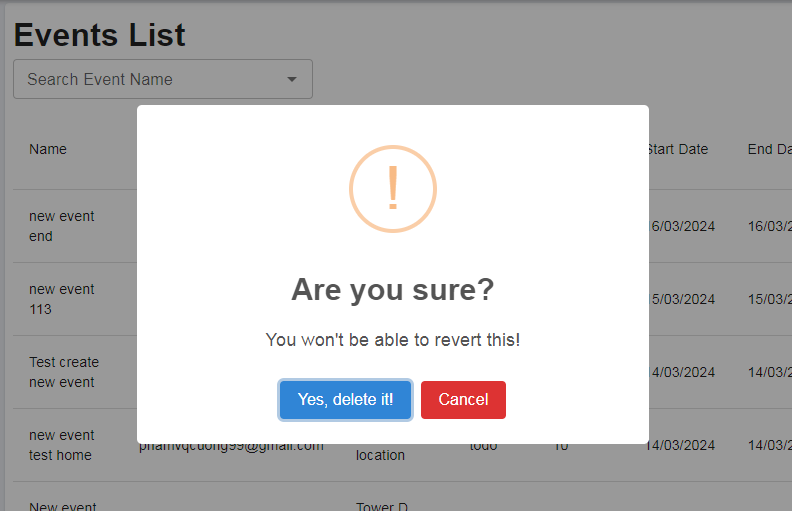


Figure 5.6 8: Pop up confirm clears the event.

# CHAPTER 6. RECOMMENDATION AND CONCLUSION

## 6.1 Conclusion:

**a. Objectives achieved:**

The website fully meets the requirements set out in the topic, including:

* Event Information Management: Manages information about department events, registration lists, statistics on the number of registrants, and attendance status tracking.
* High Integration Capability: The website can integrate with other systems such as student management, statistics, email, and reporting on past events.
* Clear Effectiveness: The website helps the department manage events professionally, saving time and costs, while improving the quality of event organization.
* Intuitive Interface: The website’s interface is scientifically designed, intuitive, easy to use, allowing users to operate quickly and efficiently.
* Diverse Features: The website provides all the necessary functions for event management such as creating, updating information, registering participation, financial management, progress tracking, etc.

**b. Limitations:**

* The website fully meets the requirements set out in the topic, including:
* Event Information Management: Manages information about department events, registration lists, statistics on the number of registrants, and attendance status tracking.
* High Integration Capability: The website can integrate with other systems such as student management, statistics, email, and reporting on past events.
* Clear Effectiveness: The website helps the department manage events professionally, saving time and costs, while improving the quality of event organization.
* Intuitive Interface: The website’s interface is scientifically designed, intuitive, easy to use, allowing users to operate quickly and efficiently.
* Diverse Features: The website provides all the necessary functions for event management such as creating, updating information, registering participation, financial management, progress tracking, etc.

## 6.2 Future Directions:

* Expand the website to manage events for the entire university.
* Develop additional functions such as allowing comments on events, managing comments.
* AI tools could support adjustments and suggestions for users when creating and editing events.
* A socket notification system could provide accurate notifications to phones.
* No Mobile Application: The website needs to develop a mobile application so that users can use it anytime, anywhere.
* Create group chats within events for students and teachers to interact and resolve queries.
* Develop events with interconnected links.

## 6.3 Summarize

The website project for managing the progress of activities and events of the Information Technology Department is a significant success, bringing many benefits to the department. The website has met the requirements and has been effective in managing the department’s events. With the outlined limitations and development directions, the website promises to continue to be perfected and developed to better meet the needs of users.

# REFERENCES

1. (“What Is React.Js?,” 2023)
2. What is React.js? Uses, Examples, & More. (2023, November 13). Retrieved March 21, 2024, from https://blog.hubspot.com/website/react-js