

Nguyen Pham Thai Son

Student ID: 001142862

Supervisor: **Doan Trung Tung**

April 2021

URL: <https://github.com/sonnpt1301/education-system>

Education System



WORD COUNT - 18340

PAGE COUNT - 150

A dissertation submitted in partial fulfilment of the University of Greenwich's BSc Computing

BSc Computing

Abstract

Learning is very important to yourself and to society. The study brings us valuable knowledge, experience, life skills that help us to orient our favorite work to contribute to the development of the country. Currently, there are many different learning methods such as studying at school, online learning under the management of teachers and school, participating in online courses on the Internet. Therefore, the goal of this project is to build a system called Education System, which is a completely online learning system, allowing users to participate in learning anywhere with smart devices that have an Internet connection. The system allows tutors to create new courses and post learning content in all fields such as Programming, Lifestyle, Business, Photograph... The system allows students to participate in courses and study completely free. In addition, the system has a feature to post blogs to help tutors or students in a course share information with each other. The activity system helps tutors to post assignment requirements and students can submit their assignments there. In addition, the system integrates real-time chat feature to help tutors and students talk to each other through text messaging.

Keywords: *education system, study online, education, study online free.*

Acknowledgements

Since this is a relatively large project for me so far and also a complete project by me. In order to be able to successfully implement this project, I am extremely thankful for the active guidance of my two lecturers, Mr. Doan Trung Tung and Mr. Do Quoc Binh, who has helped me successfully execute this project in the process. Without the devoted support of the faculty at the University of Greenwich, I would not have been able to accomplish this project effectively and in the right direction. I can't forget to remember the support of my colleagues, Nguyen Huu Quan and Nguyen Van Luong, who provided me with ideas and inspiration. They helped me a lot in times when I ran into difficulties and lacked ideas, they were also the end-user I tested the whole system after it finished. Throughout the execution of this project, I am extremely grateful for the enthusiastic support and advice provided by the teachers and friends.

Table of Contents

ABSTRACT	2
ACKNOWLEDGEMENTS	3
INTRODUCTION	12
1 LITERATURE REVIEW.....	13
1.1 EDUCATION SYSTEM DOMAIN.....	13
1.1.1 <i>Website application</i>	13
1.1.2 <i>Development of Education Website System</i>	14
1.1.3 <i>Education website system</i>	18
1.2 TECHONOLOGIES.....	19
1.2.1 <i>RESTful API</i>	19
1.2.2 <i>React</i>	20
1.2.3 <i>JavaScript (Nodejs)</i>	22
1.2.4 <i>MongoDB</i>	23
1.2.5 <i>Third party web service</i>	26
1.2.6 <i>Conclusion</i>	27
1.3 METHODOLOGY.....	29
1.3.1 <i>Waterfall</i>	30
1.3.2 <i>Conclusion</i>	31
2 REVIEW OF THE SIMILAR EDUCATION WEBSITES	32
2.1 INTRODUCTION	32
2.2 EDUCATION SYSTEM.....	32
2.2.1 <i>Udemy</i>	32
2.2.2 <i>Google Classroom</i>	37
2.2.3 <i>CMS</i>	41
2.3 CONCLUSION	47
3 REQUIREMENT ANALYSIS	48
3.1 INTRODUCTION	48
3.2 REQUIREMENT.....	48
3.2.1 <i>Course module</i>	49
3.2.2 <i>Blog module</i>	50
3.2.3 <i>Activity module</i>	51

3.3 CONCLUSION.....	56
4 SOFTWARE DESIGN.....	56
4.1 INTRODUCTION	56
4.2 CONCEPTUAL DESIGN.....	56
4.3 ENTITY RELATIONSHIP DIAGRAM	57
4.4 ACTIVITY DIAGRAM.....	63
4.4.1 <i>Create new course process</i>	63
4.4.2 <i>Create new blog process</i>	63
4.4.3 <i>Submit activity file process</i>	64
5 SOFTWARE IMPLEMENTATION	66
5.1 INTRODUCTION	66
5.2 INSTALL NODE PACKAGE MANAGEMENT (NPM).....	66
5.3 INSTALL YARN	68
5.4 INITIALIZE THE REACT APP	68
5.5 INITIALIZE BACKEND WITH EXPRESS.....	73
5.6 CREATE THE MONGODB	78
5.7 INSTALL PACKAGE	81
5.8 SET UP UPLOAD AND DOWNLOAD FILES FROM S3	82
5.9 EXAMPLE OF IMPLEMENTING THE API	84
5.9.1 <i>API create course</i>	84
5.9.2 <i>Join course function</i>	85
5.9.3 <i>Update the status for course and status when the deadline is over</i>	90
5.10 API DOCUMENT	92
6 EVALUATION AND CONCLUSION.....	96
6.1 INTRODUCTION	96
6.2 SECURITY	97
6.3 TESTING	101
6.4 EVALUATION OF EDUCATION SYSTEM	101
6.4.1 <i>Administrator</i>	101
6.4.2 <i>Tutors</i>	104
6.4.3 <i>Student</i>	108
6.5 CONCLUSION	110
6.5.1 <i>Lessons learnt</i>	110

6.5.2 <i>Problems / difficulties</i>	110
6.5.3 <i>Future improvements</i>	112
7 CONCLUSION	113
REFERENCES.....	114
APPENDIX A – PROJECT PROPOSAL.....	116
APPENDIX B – FINAL PLAN	125
APPENDIX C – EDUCATION SYSTEM FUNCTIONALS	127
APPENDIX D – EDUCATION SYTEM SCREENSHOTS	137
APPENDIX E – TEST CASE.....	146
APPENDIX F – WIREFRAMES	151

List of Figures

FIGURE 1 WEB APPLICATION VS WEBSITE.....	13
FIGURE 2 EDUCATION 1.0: LEARNERS AS RECEPTACLES OF KNOWLEDGE (BLASCHKE, ET AL., 2014)	15
FIGURE 3 EDUCATION 2.0: LEARNERS AS COMMUNICATING, CONNECTING, AND COLLABORATING (BLASCHKE, ET AL., 2014)	16
FIGURE 4 EDUCATION 3.0: LEARNERS AS CONNECTORS, CREATORS, AND CONSTRUCTIVISTS (BLASCHKE, ET AL., 2014).....	17
FIGURE 5 CLIENT-SERVER COMMUNICATION	20
FIGURE 6 FRONT-END FRAMEWORKS (ISLAM, 2019)	21
FIGURE 7 DBMS MANAGES THE INTERACTION BETWEEN THE END USER AND THE DATABASE (CARLOS, ET AL., 2011)	24
FIGURE 8 MONGODB	25
FIGURE 9 MONGODB	26
FIGURE 10 BUCKET ON S3.....	27
FIGURE 11 REACT VS VUE VS ANGULAR DOWNLOADS IN PAST 2 YEARS.....	28
FIGURE 12 DOWNLOADS PER YEAR OF FRAMEWORKS	28
FIGURE 13 TOP LANGUAGES WAS USED IN 2020	29
FIGURE 14 THE SYSTEMS DEVELOPMENT LIFE CYCLE (ALAN, ET AL., 2012, P. 11)	30
FIGURE 15 WATERFALL MODEL (ALAN, ET AL., 2012, P. 51)	30
FIGURE 16 UDEMY – HOME PAGE	33
FIGURE 17 UDEMY - COURSE DETAIL	34
FIGURE 18 UDEMY - CATEGORY	35
FIGURE 19 UDEMY - LIST COURSE	36
FIGURE 20 UDEMY - COURSE VIDEOS.....	37
FIGURE 21 GOOGLE CLASSROOM - HOME PAGE	38
FIGURE 22 GOOGLE CLASSROOM - CLASS DETAIL PAGE	38
FIGURE 23 GOOGLE CLASSROOM - CREATE CLASS.....	39
FIGURE 24 GOOGLE CLASSROOM - CLASS INFORMATION HEADER	39
FIGURE 25 GOOGLE CLASSROOM – CLASSWORK PAGE.....	40
FIGURE 26 GOOGLE CLASSROOM - PEOPLE PAGE.....	41
FIGURE 27 CMS - HOME PAGE	42
FIGURE 28 CMS - CATEGORIES	42
FIGURE 29 CMS - LIST OWN COURSE	43
FIGURE 30 CMS - JOIN COURSE	44
FIGURE 31 CMS - ACTIVITY/SHARE FILE IN COURSE	44
FIGURE 32 CMS - ACTIVITY DETAIL	45
FIGURE 33 CMS - DETAIL GRADE OF ACTIVITY.....	46

FIGURE 34 CMS - ONLINE STUDENT & CHAT	47
FIGURE 35 COURSE MODULE	49
FIGURE 36 BLOG MODULE.....	50
FIGURE 37 ACTIVITY MODULE	51
FIGURE 38 CONCEPTUAL DESIGN – EDUCATION SYSTEM.....	57
FIGURE 39 USER SCHEMA	58
FIGURE 40 CATEGORY SCHEMA	58
FIGURE 41 COURSE SCHEMA.....	59
FIGURE 42 COURSE VIDEO SCHEMA.....	60
FIGURE 43 BLOG CHEMA	60
FIGURE 44 LIKE & COMMENT SCHEMA.....	61
FIGURE 45 ACTIVITY SCHEMA.....	61
FIGURE 46 USER-COURSE SCHEMA.....	62
FIGURE 47 EDUSYS SCHEMAS	62
FIGURE 48 ACTIVITY DIAGRAM - CREATE NEW COURSE PROCESS.....	63
FIGURE 49 ACTIVITY DIAGRAM - CREATE NEW BLOG PROCESS	64
FIGURE 50 ACTIVITY DIAGRAM - SUBMIT ACTIVITY FILE PROCESS.....	65
FIGURE 51 NODEJS HOME PAGE.....	66
FIGURE 52 CHECKING WITH COMMAND NPM	67
FIGURE 53 CHECKING WITH COMMAND NPM -V AND NODE -V.....	67
FIGURE 54 NPM HOME PAGE.....	68
FIGURE 55 CREATE REACT APP	68
FIGURE 56 CREATE REACT APP SUCCESSFUL	69
FIGURE 57 FOLDER STRUCTURE OF REACT APP	69
FIGURE 58 RUNNING REACT APP	70
FIGURE 59 REDUX STORE	71
FIGURE 60 CREATE ACTION OF REDUX.....	71
FIGURE 61 CONSTANT VARIABLES	72
FIGURE 62 USER REDUCER FILE	73
FIGURE 63 INITIALIZE BACKEND	74
FIGURE 64 STRUCTURE FOLDER OF BACKEND.....	74
FIGURE 65 MIDDLEWARES	75
FIGURE 66 INITIALIZE SERVER.....	75
FIGURE 67 SETUP USE ENVIRONMENT VARIABLES.....	76
FIGURE 68 SETUP ENVIRONMENT VARIABLE	76

FIGURE 69 SETUP MAILER FUNCTION.....	77
FIGURE 70 CREATING SCHEMAS FOR USERS.....	78
FIGURE 71 CREATE THE PROJECT MONGODB	79
FIGURE 72 CREATE A CLUSTER.....	79
FIGURE 73 CREATE MONGODB SUCCESSFUL.....	81
FIGURE 74 FRONT-END - INSTALL PACKAGES.....	81
FIGURE 75 BACKEND - INSTALL PACKAGE.....	82
FIGURE 76 AWS_ACCESS_BUCKET	82
FIGURE 77 UPLOADING AND DOWNLOADING FILE FROM S3 BUCKET.....	83
FIGURE 78 API CREATE NEW COURSE	84
FIGURE 79 CREATE COURSE SUCCESS	85
FIGURE 80 GET COURSE DETAIL API	86
FIGURE 81 RECEIVE SECRET KEY OF THE COURSE THROUGH EMAIL	87
FIGURE 82 EMAIL TEMPLATE	88
FIGURE 83 SEND EMAIL SERVICE	89
FIGURE 84 JOIN COURSE API	90
FIGURE 85 NODE-SCHEDULE FOR ACTIVITY MODULE	91
FIGURE 86 NODE-SCHEDULE FOR COURSE MODULE	91
FIGURE 87 LOGIN API	92
FIGURE 88 USER APIs.....	92
FIGURE 89 COURSE APIs	92
FIGURE 90 BLOG APIs	93
FIGURE 91 ACTIVITY APIs	93
FIGURE 92 BODY OF REQUEST	94
FIGURE 93 REQUEST HEADERS	94
FIGURE 94 GENERATE ACCESS TOKEN	95
FIGURE 95 THE RESPONSE OF API CREATING NEW COURSE	96
FIGURE 96 JSON WEB TOKEN BASED ON USER INFORMATION.....	97
FIGURE 97 USER LOGIN AND SYSTEM WILL GENERATE THE JWT TOKEN	98
FIGURE 98 VERIFY JSON WEB TOKEN (JWT)	98
FIGURE 99 ENCRYPT THE USER PASSWORD	99
FIGURE 100 ENCRYPT THE USER PASSWORD	100
FIGURE 101 COMPARE PASSWORD FUNCTION.....	100
FIGURE 102 VALIDATE ROLE	100
FIGURE 103 THE FUNCTION OF ADMINISTRATOR.....	100

FIGURE 104 ACCESS DENIED.....	101
FIGURE 105 DASHBOARD PAGE.....	102
FIGURE 106 MANAGE USERS.....	102
FIGURE 107 MANAGE CATEGORIES.....	103
FIGURE 108 MANAGE COURSES	104
FIGURE 109 MANAGE COURSE	105
FIGURE 110 VIEW LIST STUDENTS IN THE COURSE	105
FIGURE 111 MANAGE STUDENT'S BLOG	105
FIGURE 112 CREATE NEW BLOG	106
FIGURE 113 COMMENT ON STUDENT'S BLOG.....	107
FIGURE 114 MANAGE ACTIVITIES IN COURSE	107
FIGURE 115 VIEW ALL STUDENT'S SUBMISSION AND GRADE	108
FIGURE 116 JOIN COURSE	108
FIGURE 117 SUBMIT FILE AND VIEW GRADE	109
FIGURE 118 CHAT SYSTEM	109
FIGURE 119 HOME PAGE	137
FIGURE 120 LIST COURSE PAGE – TUTORS	138
FIGURE 121 CREATE NEW COURSE FORM	139
FIGURE 122 EDIT COURSE INFORMATION FORM	140
FIGURE 123 COURSE DETAIL PAGE – TUTORS	141
FIGURE 124 BLOG PAGE – TUTORS	142
FIGURE 125 BLOG DETAIL	143
FIGURE 126 ACTIVITY PAGE.....	144
FIGURE 127 ACTIVITY DETAIL.....	145
FIGURE 128 LIST STUDENT'S SUBMISSION AND GRADE	145

List of Tables

TABLE 1 COMPARISON OF WEBSITE VS WEBSITE APPLICATION	14
TABLE 2 COMPARISON OF EDUCATION WEBSITES	47
TABLE 3 LIST BUGS.....	111

Introduction

The education system is an indispensable system in countries around the world today. Everyone who is born and raised must study in an educational environment with tutors in the school. Learning can help students acquire the necessary skills, knowledge, and experience so that they can orient a job in accordance with their personal strengths and interests contributes capacity to promote a more and more developed society. However, the current education system has many different approaches. Firstly, the traditional method is to study in schools under the management and teaching of the school and teachers. Secondly, the online learning method, instead of going to school and studying at school, students can learn and interact with online instructors through popular supporting software such as Zoom, Google Meet. Thirdly, with the development of the Internet, many websites have been built to provide online course content, which helps to bring more flexibility in terms of time for learning and suitable for all ages who want to participate in learning. With the method of following and learning online courses, people can join and learn by themselves at any time without having fixed study time like traditional learning methods at school. The question arises are current learning approaches really effective? From the method of taking online courses and self-learning to the traditional method managed and taught by schools and teachers, which approach will be more effective? Therefore, the project of Education System is developed, which is a combination of online learning and a mentor to manage the user's journey. The report includes 9 main parts. The first part will study the development of the education website system. The second part will explore other similar Education Systems. The third, fourth, and fifth sections will outline the processes for system development including the steps of system analysis, system design, and implementation. Section seven will evaluate the results and development of the system. Relevant documents will be attached in section eight.

1 Literature Review

1.1 Education System Domain

1.1.1 Website application

A website application is a program or application consisting of many websites with a client and server model. A Web app uses a web browser as a client to communicate and execute functions to communicate with web applications via the Internet. Web application differs from the website, website simply provides static HTML pages including text, images, and videos linked together through the links that are returned when the user clicks those links, there is no interaction. Unlike a website, a web application is a program or application that interacts between users and a web app such as logging in, registering, purchasing products, posting, and sharing information on web apps (Figure 1 Web application vs Website). Web application is developed with the aim of being able to reach more users for the purpose of individuals or businesses. Some web applications are appearing a lot on the Internet such as e-commerce web applications that help businesses manage products, track sales, and can promote product purchases to users or school management web applications that help schools manage courses, manage students of schools. There are also many other web applications such as recruitment management, restaurant management, accounting software, supermarket checkout software...

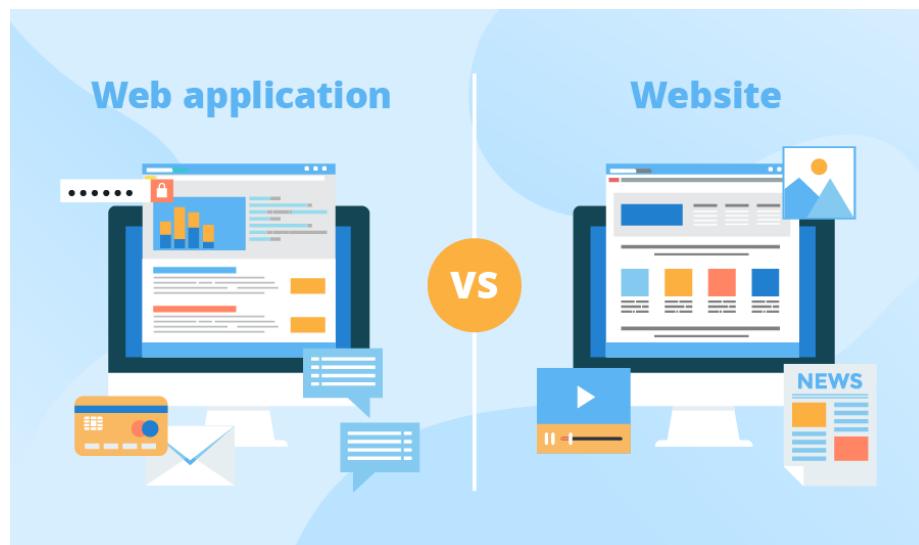


Figure 1 Web application vs Website

Source: <https://www.scnsoft.com/blog/web-application-vs-website-finally-answered>

- The difference between website and web application

Table 1 Comparision of website vs website application

Website	Web application
Low interactivity, just static HTML pages including text, images, videos... Users can only use the links to switch between pages, view, and read information.	Highly interactive, more functions such as register / login to access the application, add, edit, delete information, post, upload files, search, comment...
Is made up of static HTML pages including images, videos, plain text	It is formed by two parts client and server. Client is created by specialized languages to build the interface where the user can see it. Server is created by back-end languages (PHP, Java, JavaScript) to help web application become more interactive and perform functions in web application.
The data is simply displayed and stored statically, and the user cannot interact with it	Perform functions, interact with data

1.1.2 Development of Education Website System

In this section, the report presents a comparison of the development of the Internet in general and the development of the Education website in particular.

Education 1.0: A Pedagogical, Essentialist Education

Education 1.0 is a form of substantive education. Education 1.0 is based on three factors - student/student will absorb knowledge from teachers by listening to information transmitted from teachers, taking notes of that knowledge, studying documents, and working exercises on paper (Figure 2 Education 1.0: Learners as receptacles of knowledge . As far as mentioned by (Blaschke, et al., 2014), "Education 1.0 a standardized / one-size-fits-all education."

Education 1.0: Learners as receptacles of knowledge

Receiving
Responding
Regurgitating



For the education sector, before the Internet is present, teachers will be the holders of key knowledge and information. In addition to the material available at the library, students are still heavily dependent on information coming from their teachers. Therefore, Derek W. Keats and J. Philipp Schmidt (2007) made a comparison of the similarity between Education 1.0 and Website 1.0, which is a one-way communication process. For Education 1.0, students can only receive information from teachers, documents, textbooks, and even Website 1.0, where static websites are hosted, can only receive information.

Education 2.0: An Andragogical, Constructivist Approach to Teaching and Learning

With the rapid development of the Internet and media, Web 2.0 allows users to directly interact with the website's content, allowing users to comment and share information to other social networks, allowing users to interact with each other synchronously or asynchronously.

Just like Web 2.0, Education 2.0 also has more interaction between teachers and students, students and students. The relationship between teachers and students, students and students

contribute to accelerating learning progress. Education 2.0 focuses on three factors - communication, contribution, and cooperation (Figure 3 Education 2.0: Learners as communicating, connecting, and collaborating). With these three factors, teachers and students can interact more with each other, share information with each other and help each other in the learning process to help students progress and gain more knowledge. With the development of Web 2.0, the learning method has also been improved a lot with the presence of many tools to support the learning process such as using blogs to share information with each other social networking sites, podcasts ... Although the learning method has been greatly improved compared to Education 1.0, Education 2.0 transmits knowledge and controls the classrooms is still teachers.

Education 2.0: Learners as communicating, connecting, collaborating



Figure 3 Education 2.0: Learners as communicating, connecting, and collaborating (Blaschke, et al., 2014)

Education 3.0: A Heutagogical, Connectivist Approach to Teaching and Learning

With the development of Web 3.0, it allows users to have a richer experience with a unique personalization for each individual using the Web. Education 3.0 is also changing with a more liberal education of self-selecting learning content as well as creating learning content and

sharing them on the Internet or social networking sites. Education 3.0 broke the boundaries between teachers and students, students and students. Education 3.0 focuses on three elements - connector, creator, and constructivist (Figure 4 Education 3.0: Learners as connectors, creators, and constructivists . Unlike Education 1.0 and Education 2.0, Education 3.0 allows students or any object to decide for themselves in the choice of learning content instead of having someone to control and transmit knowledge. Education 3.0 focuses on individual self-study based on each person's interests and goals. The transformation of Education 3.0 makes learners possible to become content creators, learners, classroom administrators, and instructors. Now teachers are just the ones acting as instructors, suggestions, and orientations for students about the direction and way of learning effectively.



Figure 4 Education 3.0: Learners as connectors, creators, and constructivists (Blaschke, et al., 2014)

1.1.3 Education website system

Application or website application developed in education today has appeared a lot on the Internet, including many different forms with many different purposes such as teaching, school

management... With the continuous development of modern technology as well as the Internet, all ages from students to adult own laptops or smartphones, so the strongly developed education website application is an alternative solution for traditional education method. With an educational website application, schools can easily manage all student information, courses, students can keep track of class schedules, study results, and important announcements through the application. Everything will be managed through the app and accessible anywhere with your laptop or smartphone with Internet access. In addition, educational website application solved many limitations for the traditional method before.

Important notices are notified through the website application or email provided to all students so that all students can capture all the latest information, the latest notice they need to capture. In the traditional method, the information is only transmitted through verbal communication, which leads many students to assume that they have not heard the message.

Students can track their grades or results directly through the app anywhere. Compared with the traditional method, students' tests or scores are often saved in paper form, and students or teachers may lose them. In addition, using website application when students view their score results on the website application, they help protect their privacy.

Communication with teachers: Compared with the traditional method, this method is quite limited when students have to meet their teacher directly to discuss questions or some private issues. Website application communicates via chat dialog or via email allowing them to talk anywhere with higher privacy protection...

Anti-cheating in exams: With the marking system for teachers will be able to help them prevent exam cheating among students with the traditional method of writing on paper and pen. In addition, reduce the time spent marking students and avoid losing students' papers by storing too many paper tests with traditional methods.

In addition to the school management education systems, many educational website systems also appear many websites with content on courses posted by universities or individuals with knowledge of a field share some fields. These systems allow users to access the website and participate in or buy premium courses and study online directly on the system. The systems

provided to users cover many different fields such as Marketing, Business, Development, Lifestyle ... to help learners not be limited by genres. Courses include free and paid courses, for free courses, users can participate and learn completely free, for courses including paid, users will have to is required to pay for the course to be able to take the course. With online learning, users will not be limited by the time to participate in learning, there is no age limit to participate, do not have to move anywhere, just have a smart device with an internet connection.

1.2 Techonologies

1.2.1 RESTful API

The server side of the Education System uses RESTful API, which is a popular standard in designing APIs for web applications to help clients and servers communicate with each other through the client-server model. Instead of using the server-side rendering model, most of the processing logic is done on the server-side from the basic logic of validation, reading data to complex authorization, payment, conversion logic, current login. display the data to display on the web page. In a nutshell, whenever a user accesses the website, the browser sends a request to the server, the server that receives the request will retrieve it from the database and render the HTML and return it to the browser to display to the user.

Because all data logic processing is on the server-side, server-side rendering has some advantages such as fast initial speed, ease to optimize, easier to understand, and easier of code, so the developer only needs a single project. There is no need to separate the front-end and the back-end. However, there is a noticeable disadvantage of server-side rendering that makes the user experience not really good due to having to load the page too many times every time a user makes a page redirect. Because the server-side has to process logic, data and have to spend a lot of redundant and duplicate data including HTML, header, footer, this leads to server overload and bandwidth consumption.

Client-side rendering is a model created to overcome the weaknesses of the server-side rendering model with the SPA (Single Page Application) mechanism, the application is only located on one page. This model from logic validation, data reading, sorting, filtering, page conversion logic, rendering for display on the browser are all handled by the client-side. However,

complex logic such as payment, authorization ... still needs to be handled on the server-side. Some advantages of the client-side rendering model, the page only needs to load only the first time when the user visits the page when the user sends the request to get data, the user can see the data. New document without loading the page. Offloading server-side processing work due to some logic being done on the client-side, minimizing bandwidth overhead by just getting JSON and required data instead of having to render the entire page again like the server-side rendering model. The client-side rendering model is used with websites with a lot of user interaction, SPA will make the user experience better when the user does not need to reload the page many times.

However, in the Education System project use RESTful API, which is the bridge, communication method between client and server. Therefore, the Education System will use a client-server model to reduce processing workload on both the client and server side. The client side will focus on processing data on the return server side, the server will handle the logic of data, login, authorization ... The client side will send requests and receive responses from the server via axios or fetch to call the api needed. The server-side APIs will receive the corresponding input parameters and return the data so that the client side can receive it each time a request is sent.



Figure 5 Client-Server communication

1.2.2 React

Due to the use of the client-server model, it is necessary to separate front-end and back-end and to save time and be able to develop quickly, the project will use existing frameworks to build the user interface. Nowadays, there are various frameworks have been developed to assist developers in building the client-side (user interface). Frameworks provide frameworks, UI components are available such as menus, navbars, buttons, forms... to help developers easily and reduce the time when building user interfaces attracts users and compatibility with all popular browser platforms such as Chrome, Internet Explorer. Besides, these frameworks provide general documentation, common paradigms hence developers can easily learn about them and can easily

find answers to common problems for them. Some of the most popular frameworks currently used by developers are React, Angular, and Vue.

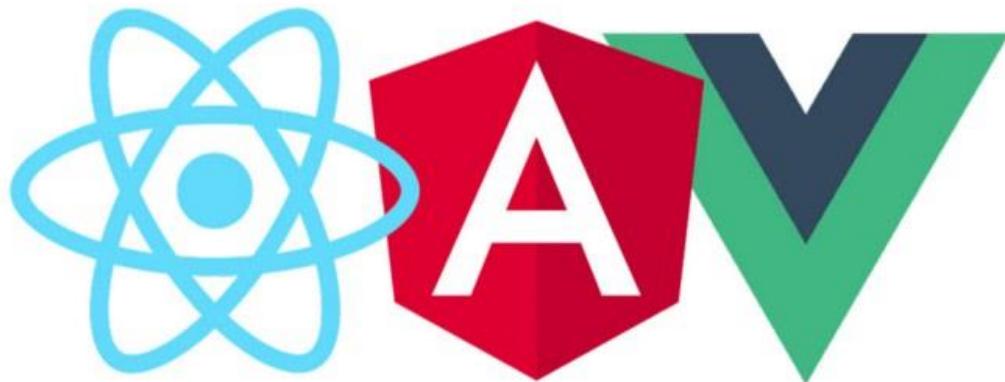


Figure 6 Front-end Frameworks (Islam, 2019)

React is a well-known JavaScript library used for frontend development, was developed by Facebook. React is developed in the direction of Single Page Application (SPA) to deal with large-scale systematic websites and large amounts of data (Alex & Eve, 2017). The main purpose of React is to make the website operate faster and more efficiently by splitting the components in the website and managing them separately with separate states. These small components can only be changed if their state is changed, which will make the user experience better when the user changes the state of a component, only that component will be re-render instead of reloading the entire website.

➤ **Advantages of React**

- Virtual DOM: Because the real DOM of a website is very large, when updating or changing a small part of the page will cause the website to reload completely, which takes a long time to reload and slows down the whole system of the website, this will do reduce performance and user experience when unrelated components are reloaded. Using the virtual DOM to interact with the real Dom will help the website have better performance

and user experience when the virtual DOM will be calculated when there is any change or update before brought to the real DOM.

- Reusable Components: React supports splitting the content of the page into small components, so components that appear multiple times can be reused instead of over and over.
- Community: Since Facebook has been developing React into open source since 2013, the community using React is huge. From there, developers can search for answers to common problems encountered on sites like Stack Overflow, Facebook group ...
- Maintain: Regularly maintained and updated by Facebook.
- React Developer Tools: is an extension used on Chrome or Firefox browsers to assist developers in the process of interface building and debugging.

➤ **Disadvantages of React**

- React was developed under Single Page Application, so it is not SEO friendly
- To be able to build large and complex user interfaces, developers also have to use many other related libraries such as Redux, React Hook, React Router Dom...

1.2.3 JavaScript (Nodejs)

The front-end is the client side, also known as the user interface, where the user can see how the interface is arranged and displayed. But if only the front-end exists, users cannot perform operations with the website such as manipulating data. Therefore, a complete website needs to have a back-end, which handles logic to be able to return data to the user when the client-side sends the request to the server, the server will receive the request from the client and return the results based on the request on the client-side and the user cannot see how the content of the back-end is handled. Currently, there are many choices of languages to develop on the popular back-end (server-side) such as JavaScript, PHP, Ruby, Java...

JavaScript is a programming language that was developed by Brendan Eich, co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla corporation at Netscape Communications Company with the first name Mocha, then renamed LiveScript, and finally officially renamed until now to JavaScript. JavaScript is commonly used to develop both front-

end (UI) and server-side (Node Js). Since the development of the language, initially, Js was only able to develop towards the user interface until 2009 when Ryan Dahl released Node.js and use Js to run on the server-side. (Ethan, 2016)

➤ **Advantages of JavaScript**

- Can be used for building both client-side and server-side
- Easy to learn and approach if you are already using JavaScript to build the frontend, using JavaScript to build backend helps beginners do not spend too much time to learn and approach this language.
- Large community: Due to the large community using JavaScript, it is easy for developers to find answers to common problems they face. Figure 2 shows the number of JavaScript users building the backend in 2017 as reported by Stack Overflow.
- Materials and courses are widely available on the Internet to make it easier for beginners to access.

➤ **Disadvantages of JavaScript**

- Difficult in developing large-scale systems, the possible alternative is TypeScript.
- Because JavaScript is run on the client, the user is very vulnerable to attack either if the user disables JavaScript or during the process, an error occurs with the JavaScript the system stops rendering the whole system.

1.2.4 MongoDB

When building a website system, the data used in the website is extremely large for large and small websites such as user information, product information, sales information for e-commerce websites or student information, student scores with school administration websites. Therefore, in order to be able to manage all user data or other data related to the field in which the website is operating, database is a term that is a collection of data that is related to each other and is stored in a database management system (DBMS). This data is usually saved as a table or as a JSON providing the ability for developers to retrieve the data, modify it or add new data. Therefore, a website that can be active and be interactive with end users requires a database. In fact, not only in the website programming industry, there is the concept of database,

this concept exists in our real life like contacts is a form of database, helping users to reduce minimizing having to remember all contact information of relatives. Contacts allows users to store information of their relatives, add information, and query information for the contact person in the contacts to contact. Currently, there are many popular Database Management System (DBMS) can manage data such as MySQL, MongoDB, PostgreSQL...

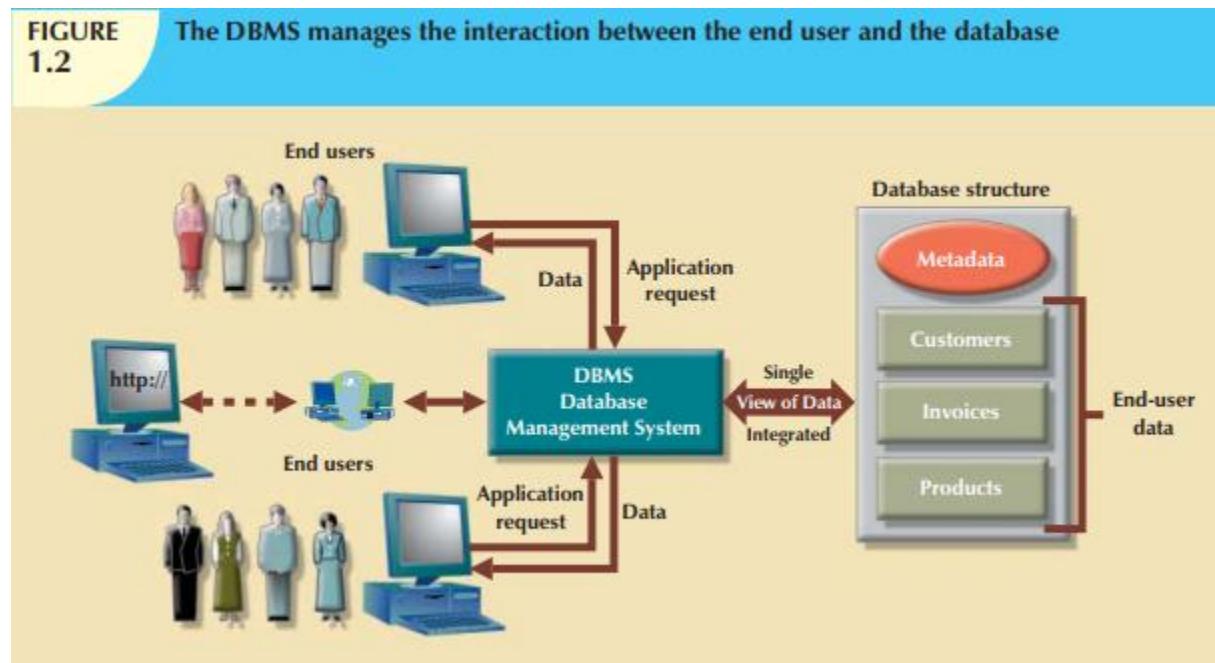


Figure 7 DBMS manages the interaction between the end user and the database (Carlos, et al., 2011)

MongoDB was developed in 2007 as a part of Platform as a Service (PaaS) by MongoDB Inc, a non-relational database, so using MongoDB makes it easier for developers to scale (Kristina, 2013). MongoDB is a document-oriented database form that is not bound by primary and subkey and is stored as JSON data type consisting of key and value pairs (Figure 8 MongoDB & Figure 9 MongoDB). Therefore, adding and deleting data for MongoDB will be easier than database whose relationship is bound by primary key and sub key. Furthermore, relational databases force developers to understand and adhere to the normalized rules. Due to its high flexibility, no ties, and high expansion of province, MongoDB is preferred for large-scale projects. Besides, MongoDB is often used for website systems with real time systems that require fast response speed, systems with big data that need high query speeds...



Figure 8 MongoDB

Source: mongodb.com

The screenshot shows the MongoDB interface with a list of documents in the 'test.books' collection. The table headers are:

	DOCUMENTS 5	TOTAL SIZE 345B	AVG. SIZE 69B	INDEXES 1	TOTAL SIZE 32.0KB	AVG. SIZE 32.0KB
Documents				Indexes		
Aggregations				Validation		
Schema						
Explain Plan						
OPTIONS				FIND		RESET
ADD DATA						

The documents listed are:

- Document 1: _id: 7000, author: "Homer", copies: 10
- Document 2: _id: 7020, title: "Iliad", author: "Homer", copies: 10
- Document 3: _id: 8645, title: "Eclogues", author: "Dante", copies: 2

Figure 9 MongoDB

Source: mongodb.com

➤ **Advantages of MongoDB**

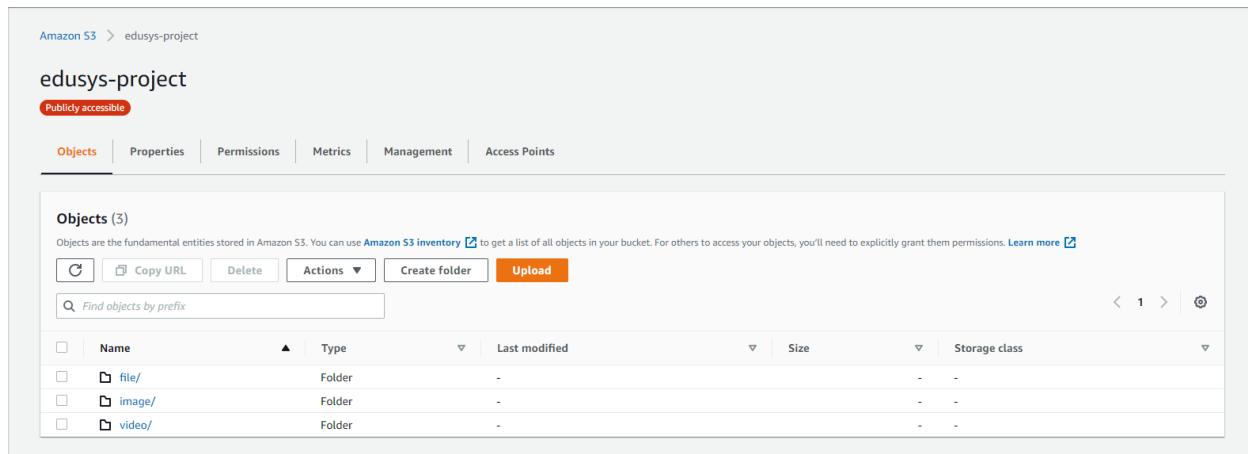
- Highly flexible, scalable due to not being bound by primary and secondary keys
- Flexible to add, edit, and delete without affecting the application
- High data query speed

➤ **Disadvantages of MongoDB**

- Since MongoDB is a non-relational database, related data can be skewed and mismatched when performing add, edit, or delete queries
- Uses a lot of memory
- Does not support auto join tables, join manually in the code process
- Limit on capacity and connection

1.2.5 Third party web service

In addition to storing user data, data information objects in the database. The Education System uses Amazon S3, a storage service provided by Amazon Web Services, to store files or images uploaded by students or tutors. The system uses a third party to back up files in order to avoid storing these files on the server, helping the server avoid increasing the size and reducing the performance of the server. The system uses S3 storage as file requirements, file assignments, post images, individual images...



The screenshot shows the Amazon S3 console interface for the 'edusys-project' bucket. The top navigation bar shows 'Amazon S3 > edusys-project'. Below the navigation, it says 'edusys-project' and 'Publicly accessible'. A toolbar at the top includes 'Objects' (highlighted in orange), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below the toolbar, there's a search bar with 'Find objects by prefix' and a pagination area with '< 1 >'. The main content area is titled 'Objects (3)'. It displays a table with columns: Name, Type, Last modified, Size, and Storage class. The table shows three entries: 'file/' (Folder), 'image/' (Folder), and 'video/' (Folder), all of which are '-' in the last three columns.

Name	Type	Last modified	Size	Storage class
file/	Folder	-	-	-
image/	Folder	-	-	-
video/	Folder	-	-	-

Figure 10 Bucket on s3

1.2.6 Conclusion

Frontend: Base on React popularity and outstanding advantages such as Virtual DOM helps web applications have better performance and user experience when calculating virtual DOM changes before interacting with the real DOM. In addition, with the growing React community, rich learning materials make it easier to approach the language, and always maintained and upgraded by Facebook, the largest social networking site today. The creator of React is using the framework itself.

The Education System project will use React to build the user interface, help reduce the time to develop the interface for the web application, and can spend a lot of time building functionality for this web application.

In addition, the number of people who downloaded and used the React framework to build their web application in the past 2 years ago and the number of downloads within 1 year compared to other frameworks shows that the developer community using React is huge. (Figure 11 React vs Vue vs Angular downloads in past 2 years, Figure 12 Downloads per year of frameworks).

In addition to the objective reasons based on the outstanding characteristics and the number of people using React, this project applies this framework to other personal reasons such as having some experience dealing with React from previous projects and experience from previous jobs at the company, so adopting React makes it easier to access and solve problems faster.

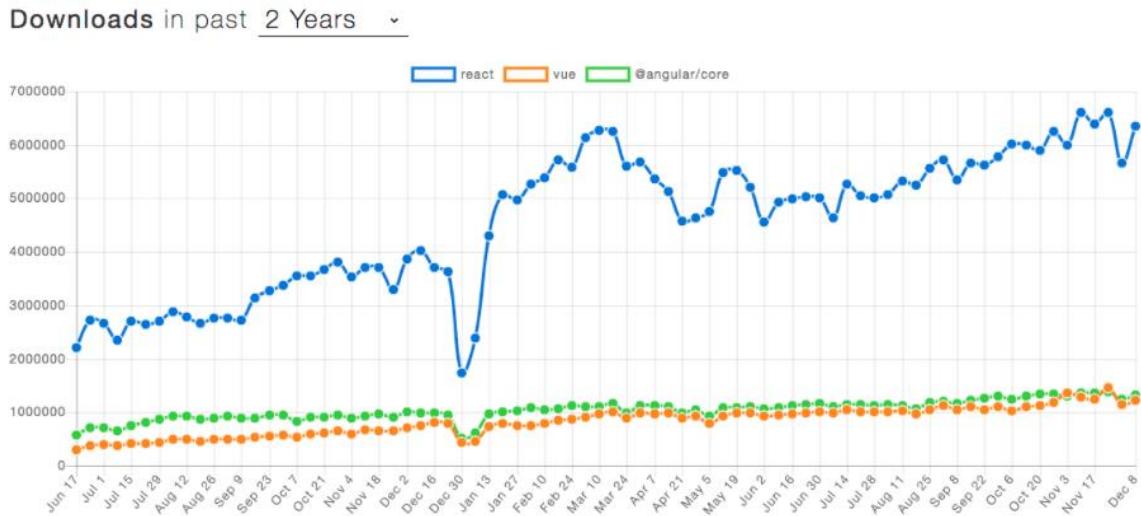


Figure 11 React vs Vue vs Angular downloads in past 2 years

Source: <https://qist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190>

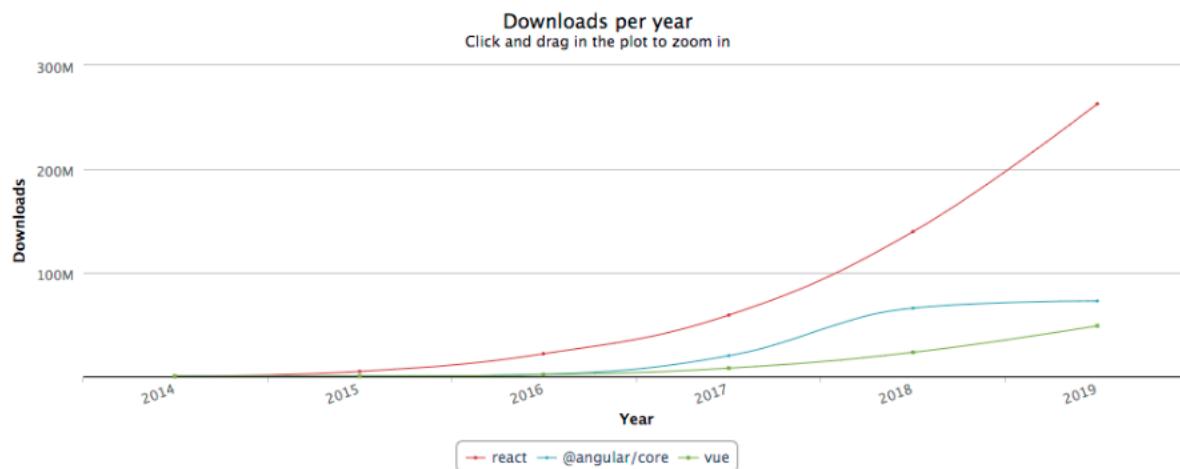


Figure 12 Downloads per year of frameworks

Source: <https://qist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190>

Programming, Scripting, and Markup Languages

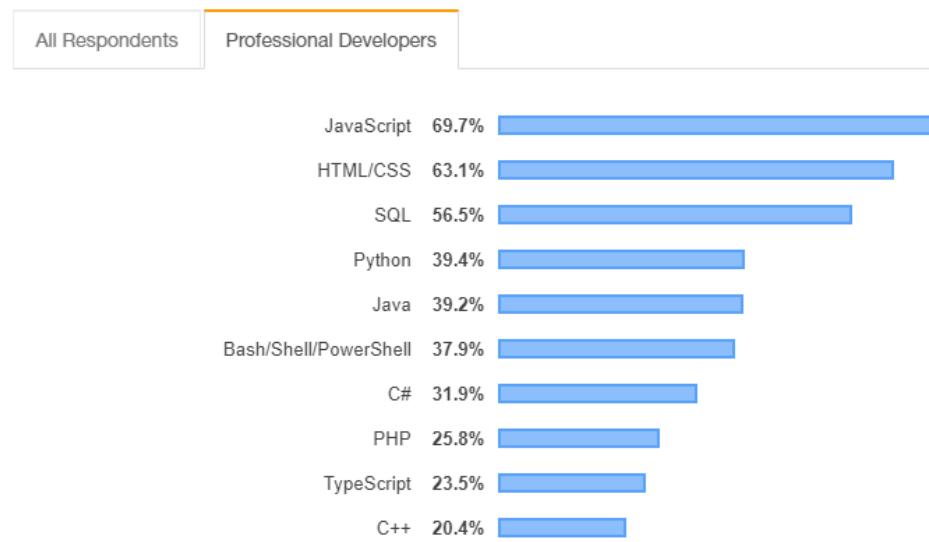


Figure 13 Top languages was used in 2020

Source: <https://codeburst.io/javascript-trends-in-2020-b194bebc5ef8>

Backend: Statistically, JavaScript is used to outperforming other languages (Figure 13 Top languages was used in 2020). Due to its large amount of usage, JavaScript has a large community that makes it easier to find answers to common problems when learning about the language...

Due to the lack of time for me to complete this project, in order to minimize the time to design the schema for the database, MongoDB will be used as the database for the School Management System. Due to the nature of MongoDB is a non-relational database and is stored as key and value pair. MongoDB schema will be created during the coding process. In addition, MongoDB is not bound by the relationship between the primary key and the subkey, thus helping to expand and avoid errors in the process of adding, modifying, or deleting data. The data query language is simpler than MySQL and has higher performance than many other databases.

1.3 Methodology

To be able to develop a project, software, or website system, it is necessary to follow a certain process to ensure the end result is a successful product, reducing time and spending and the most error. Developers must know the Software Development Life Cycle - SDLC. SDLC is a set of processes associated with a software development process that includes a plan that describes in detail what each process needs to do in order to be developed, maintained, modified, or enhanced. software level. Currently, there are many types of software development models for each project purpose or specific models for business models such as Waterfall, V-model, Spiral model, Agile ...

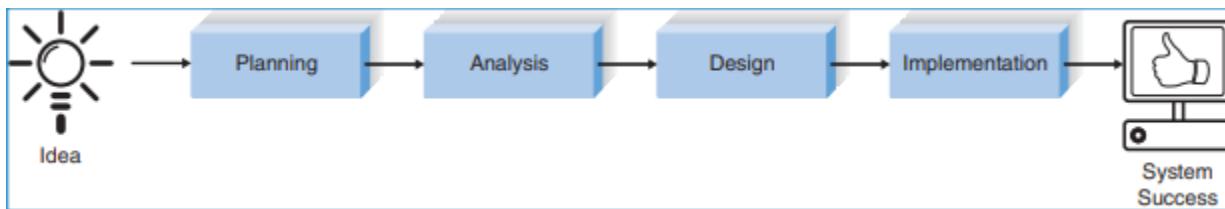


Figure 14 The Systems Development Life Cycle (Alan, et al., 2012, p. 11)

1.3.1 Waterfall

The waterfall model is the oldest and most traditional development life cycle model, the steps are performed sequentially from top to bottom as the name suggests of this waterfall model. The stages in this step will be done one by one, when the first step is finished and

moderated will move on to the next stage until the end of the development lifecycle, the following stages will be based on information from the previous period.

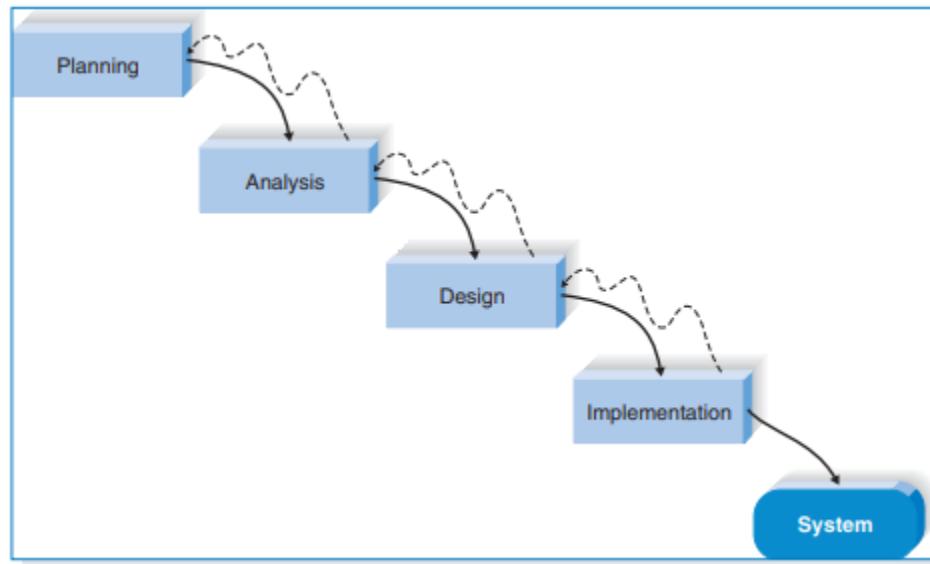


Figure 15 Waterfall model (Alan, et al., 2012, p. 51)

- **Planning:** This phase defines requirements related to functions and interfaces that the system should have. This phase needs to talk with the end-user of the product to be able to come up with the necessary requirements and fully for the product.
- **Analysis:** After gathering enough requests from customers, the developers will analyze how they can meet those needs of customers.
- **Design:** Design sketches for the system from the analysis from the requirements analysis phase
- **Implementation:** Execute the functions required in the system that has been previously analyzed and designed. After each completed function will need to be tested.
- **System and Maintain:** Stage of installation and guide to the customer how to use the system, let the customer use the product and fix the errors if any, or necessary changes that the customer requires to the system.

➤ **Advantages of Waterfall**

- Suitable for small projects, requirements are clearly defined from the outset and do not change too much.

- The stages are clearly divided, each stage needs to be completed before moving to the next stage.
- Minimize the bugs that arise in the final stage and deliver the final product because each stage has been completed and thoroughly tested to move the stage

➤ **Disadvantages of Waterfall**

- The arising errors are only corrected in a specific period.
- Not suitable for large projects and requirements be changed frequently

1.3.2 Conclusion

The waterfall model will be a development cycle model applied to the development of the Education System. Because Waterfall is a traditional and long-standing model that is easy to apply and the Education System project is expected to fit this model due to its compatibility with this model. The project will be individual implementation, requirements are also clearly planned from the beginning so the requirements may not be changed too much from the beginning of the plan. The project will be implemented step by step according to the model's phases, ending one phase will move to the next stage. That will help the project to minimize errors and minimize additional extra time.

2 Review of the similar Education Websites

2.1 Introduction

With the rapid development of technology, innovation in learning modes. Therefore, at present, there are many websites with the educational and learning background which have been applied by many schools. Not only schools apply website models to manage the school, but there are also many similar website models that are built to provide free and paid courses to improve the self-study of each student or any age group. Before developing a similar system in this project, this research will explore similar systems in terms of strengths and weaknesses. From there, at the end of this section, there will be some criteria based on similar systems and some additional criteria to improve the quality of the educational and learning foundation website model.

2.2 Education system

2.2.1 Udemy

Udemy is one of the famous websites for providing courses including free and paid courses in many different disciplines such as Design, Development, IT & Software, Music... Visit the website at <https://www.udemy.com/>

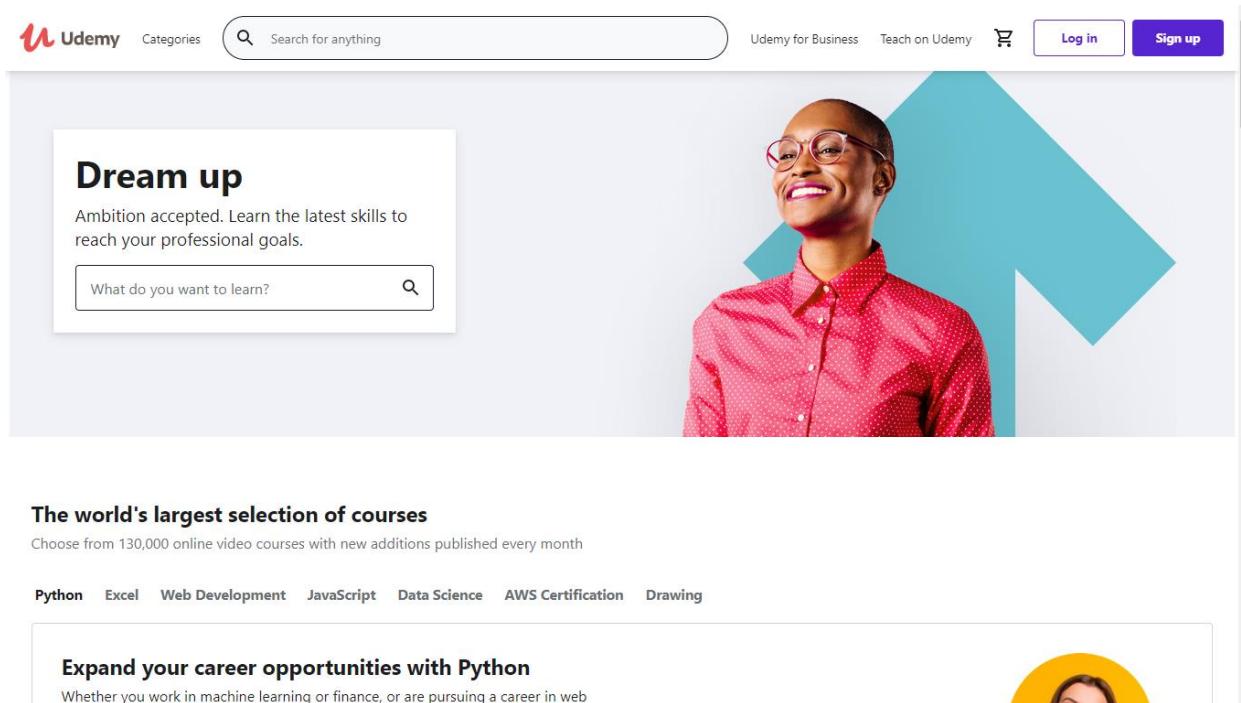
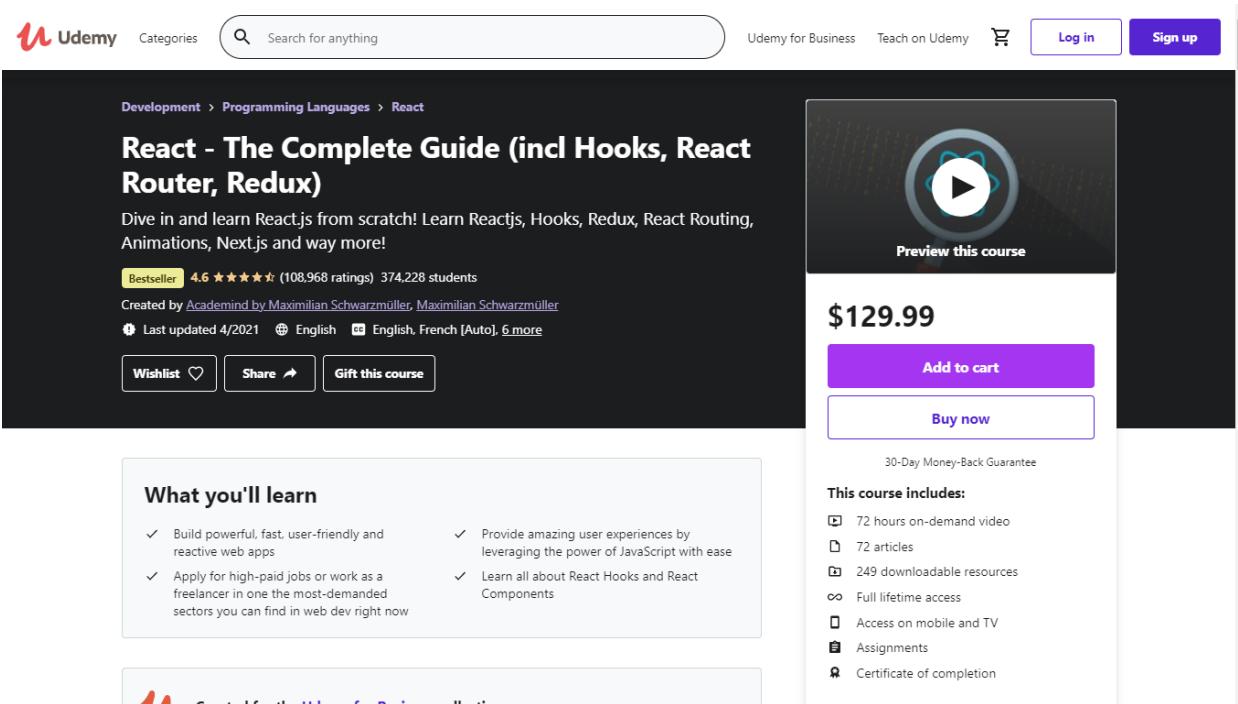


Figure 16 Udemy – Home page

Udemy is an online learning website founded by Gagan Biyani, Oktay Caglar, Eren Bali in February 2010. Udemy is a website that specializes in E-learning, online education systems, and online marketing.



The screenshot shows a course detail page on the Udemy website. At the top, there's a navigation bar with the Udemy logo, categories, a search bar, and links for 'Udemy for Business', 'Teach on Udemy', a shopping cart icon, 'Log in', and 'Sign up'.

The main content area displays the course title: 'React - The Complete Guide (incl Hooks, React Router, Redux)'. Below the title is a brief description: 'Dive in and learn React.js from scratch! Learn Reactjs, Hooks, Redux, React Routing, Animations, Next.js and way more!'. The course has a rating of 4.6 stars from 108,968 reviews and 374,228 students.

Details about the course creator ('Academind by Maximilian Schwarzmüller'), the last update date (4/2021), and language options (English, English, French [Auto], 6 more) are also shown. There are buttons for 'Wishlist' (with a heart icon), 'Share' (with a link icon), and 'Gift this course'.

A large video thumbnail with a play button is displayed, along with a 'Preview this course' button. The price of the course is listed as '\$129.99'. Below the price are two prominent buttons: 'Add to cart' (purple) and 'Buy now' (white).

On the right side, there's a '30-Day Money-Back Guarantee' section and a 'This course includes:' list. The included items are: 72 hours on-demand video, 72 articles, 249 downloadable resources, Full lifetime access, Access on mobile and TV, Assignments, and a Certificate of completion.

At the bottom left, there's a note: 'Curated for the Udemy for Business collection'.

Figure 17 Udemy - Course Detail

Here users can buy courses or join free courses and study completely online at the Udemy website. Instead of going to school to study today, online learning is a form of learning that is not bound by age, an active way to be able to learn and gain more knowledge online. With the current epidemic Covid-19 extremely serious, moving to school is not really recommended because the risk of transmission is very high. Therefore, learning online on website systems like Udemy is a really good and effective method. Students can study anywhere, study at any time of the day.

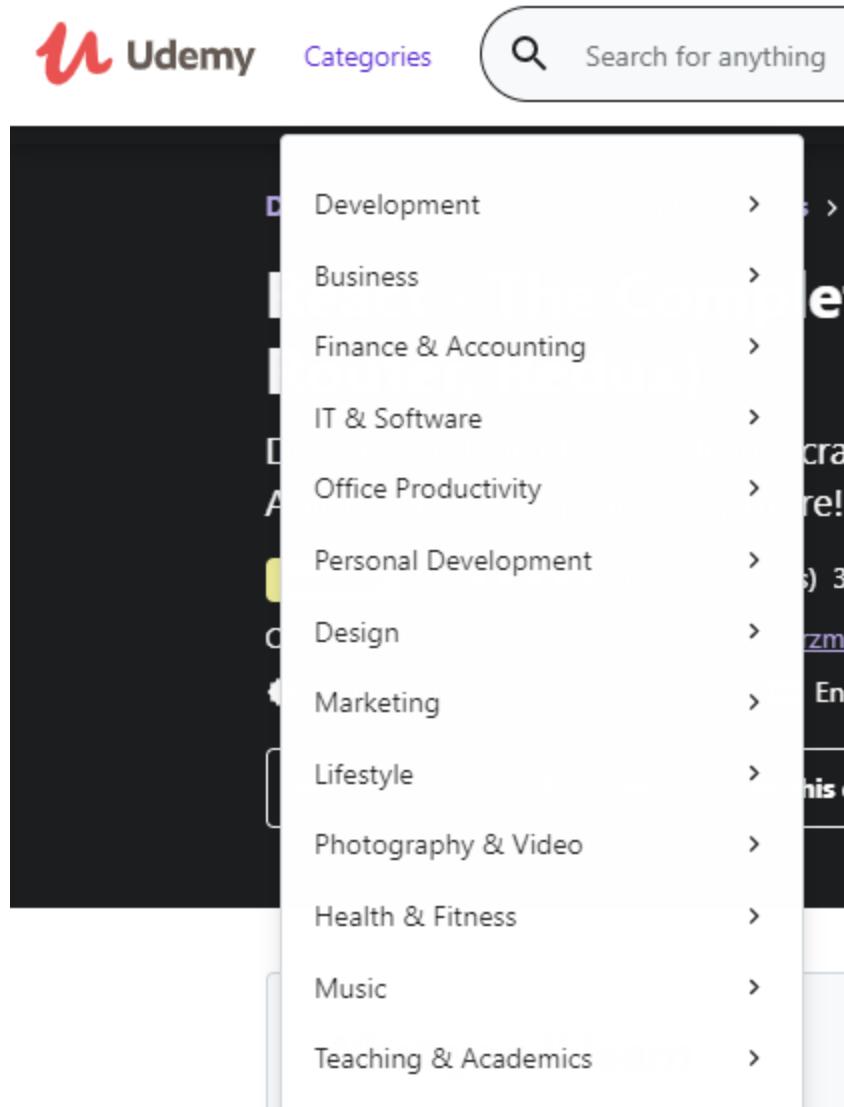
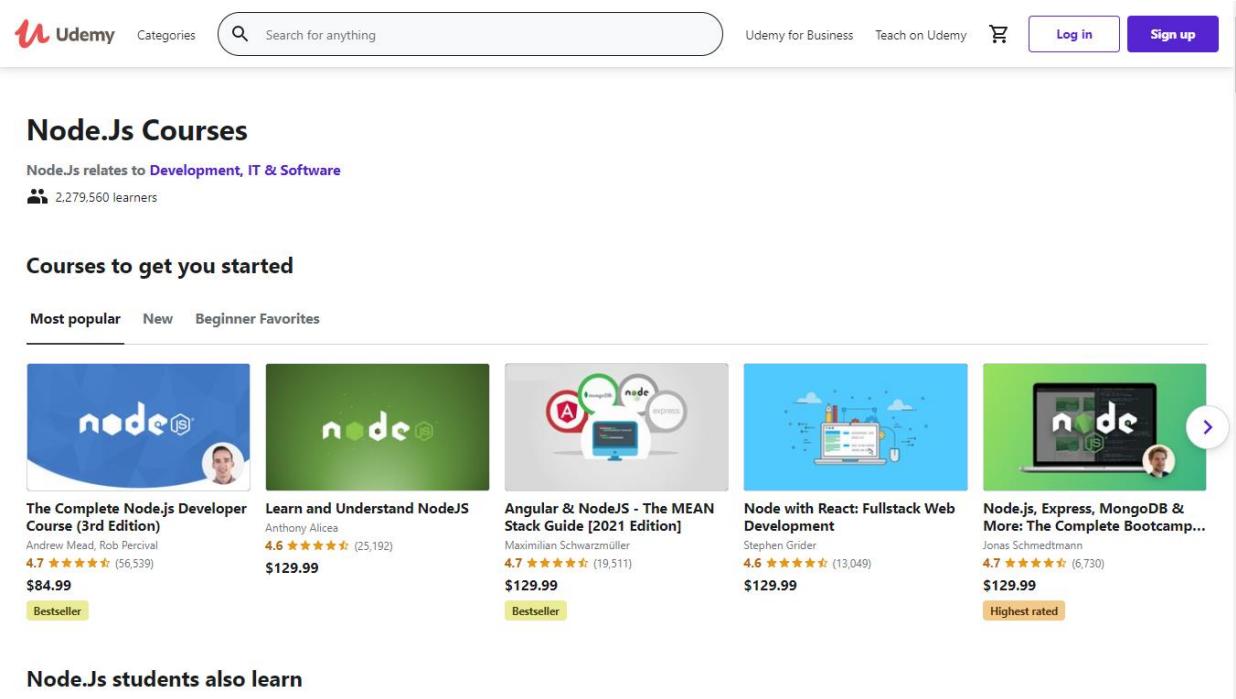


Figure 18 Udemy - Category

Udemy has many categories for users to choose suitable for the industry they are aiming for including Development, Business, IT & Software, Music...



The screenshot shows the Udemy website's search results for "Node.Js Courses". At the top, there is a navigation bar with links for "Udemy for Business", "Teach on Udemy", a shopping cart icon, "Log in", and "Sign up". Below the navigation is a search bar with the placeholder "Search for anything". The main content area is titled "Node.Js Courses" and includes a sub-section "Courses to get you started". There are five course cards displayed:

- The Complete Node.js Developer Course (3rd Edition)** by Andrew Mead, Rob Percival. Rating: 4.7 stars (56,539 reviews). Price: \$84.99. Tag: Bestseller.
- Learn and Understand NodeJS** by Anthony Alices. Rating: 4.6 stars (25,192 reviews). Price: \$129.99.
- Angular & NodeJS - The MEAN Stack Guide [2021 Edition]** by Maximilian Schwarzmüller. Rating: 4.7 stars (19,511 reviews). Price: \$129.99. Tag: Bestseller.
- Node with React: Fullstack Web Development** by Stephen Grider. Rating: 4.6 stars (13,049 reviews). Price: \$129.99.
- Node.js, Express, MongoDB & More: The Complete Bootcamp...** by Jonas Schmedtmann. Rating: 4.7 stars (6,730 reviews). Price: \$129.99. Tag: Highest rated.

Below the course cards, there is a section titled "Node.Js students also learn" which is currently collapsed.

Figure 19 Udemy - List Course

In a specific Development area like Nodejs, Udemy will offer users suggestions for the selection of courses for beginners or popular courses that many people are following and learning.

Course content

18 sections • 177 lectures • 35h 3m total length

[Expand all sections](#)

<p>^ Welcome</p>	2 lectures • 8min
<p>► Welcome to the Class! ▾</p>	Preview 06:58
<p>► Grab the PDF Guide ▾</p>	01:04
<p>^ Installing and Exploring Node.js</p>	5 lectures • 48min
<p>► Section Intro: Installing and Exploring Node.js ▾</p>	Preview 00:56
<p>► Installing Node.js and Visual Studio Code ▾</p>	Preview 08:51
<p>► What is Node.js? ▾</p>	Preview 15:31
<p>► Why Should I Use Node.js? ▾</p>	Preview 16:24
<p>► Your First Node.js Script ▾</p>	Preview 06:19

Figure 20 Udemy - Course Videos

In each course, the videos are clearly divided into each category from basic to advanced steps to help users easily follow and learn. However, most of all Udemy courses require users to pay for each particular course, the price of each course depends on the difficulty of that course. In addition, Udemy will also have promotions offering free courses to users.

2.2.2 Google Classroom

Google Classroom is a product developed by Google as an online classroom software, where teachers can open class and manage classes easily and conveniently. Visit the website at <https://classroom.google.com>

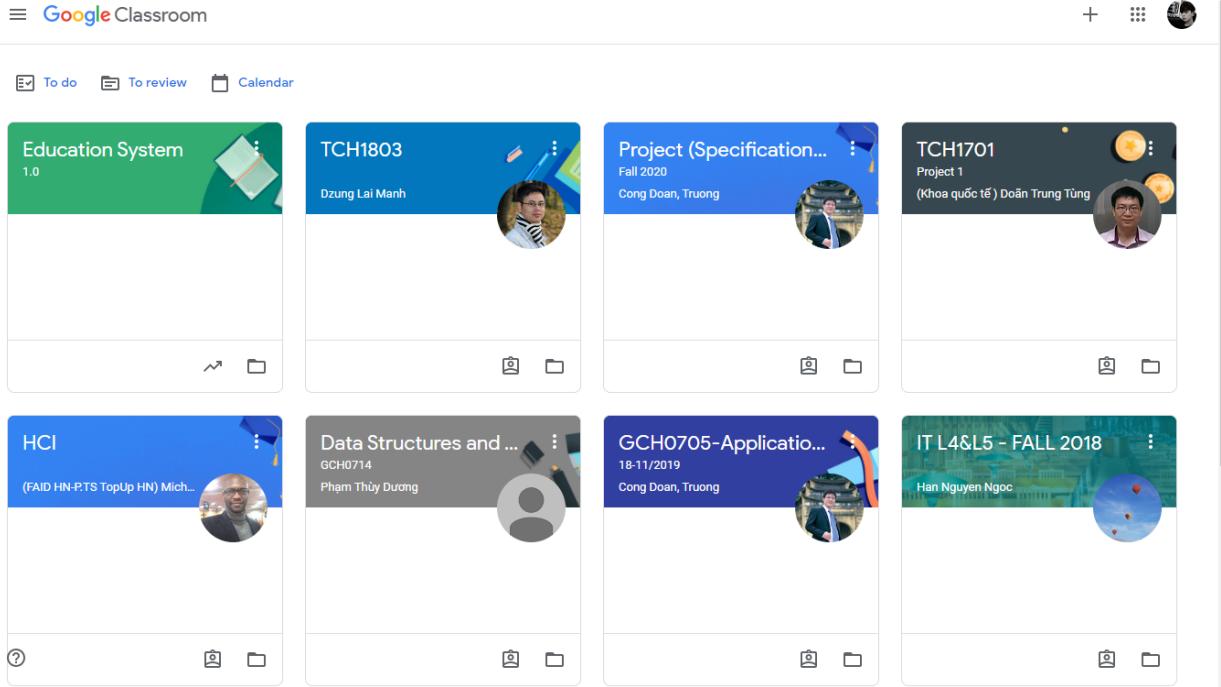


Figure 21 Google Classroom - Home page

At the home page, instructors or students can create classes or join classes. In each class, the instructor can post lectures, creating activities for all students in the class to join.

The screenshot shows the Google Classroom class detail page for the HCI class. The Stream tab is selected. The main area features a blue graduation cap icon. Below it, there's an "Upcoming" section with a message from the instructor: "Woohoo, no work due in soon!" and a "View all" button. To the right, there's a "Announce something to your class" button. A message from the instructor dated 9 Nov 2020 states: "NOTICE!!! DO NOT SEND ME PDF DOCUMENTS TO REVIEW : ONLY WORDS DOCUMENTS ARE ALLOWED BY ME. Your report Must connect with the following criteria: For your guide follow the following below and make sure they align to your coursework. Appropriate referencing in Harvard style of relevant background literature (journal papers, conference papers, academic books) throughout the report." There are also "Stream", "Classwork", and "People" tabs at the top of the sidebar.

Figure 22 Google Classroom - Class detail page

Create class

Class name (required)

Section

Subject

Room

Cancel Create

Figure 23 Google Classroom - Create Class

When the teacher creates a class, the class will have a class code generated, when a student wants to join the class will have to enter the code that the teacher provides.

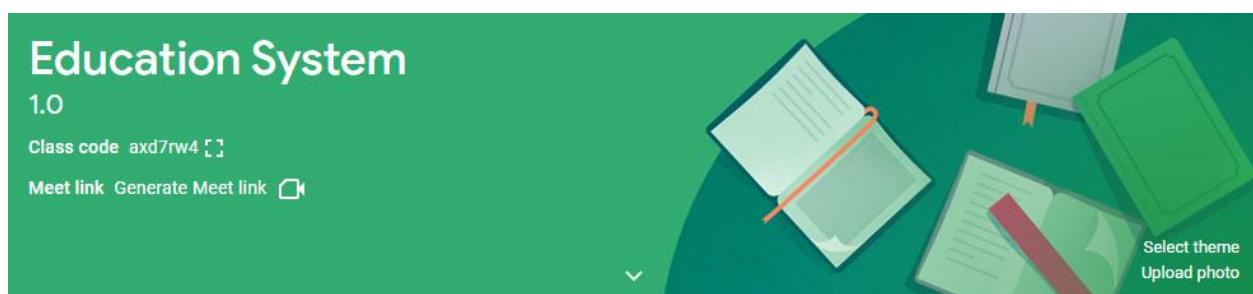
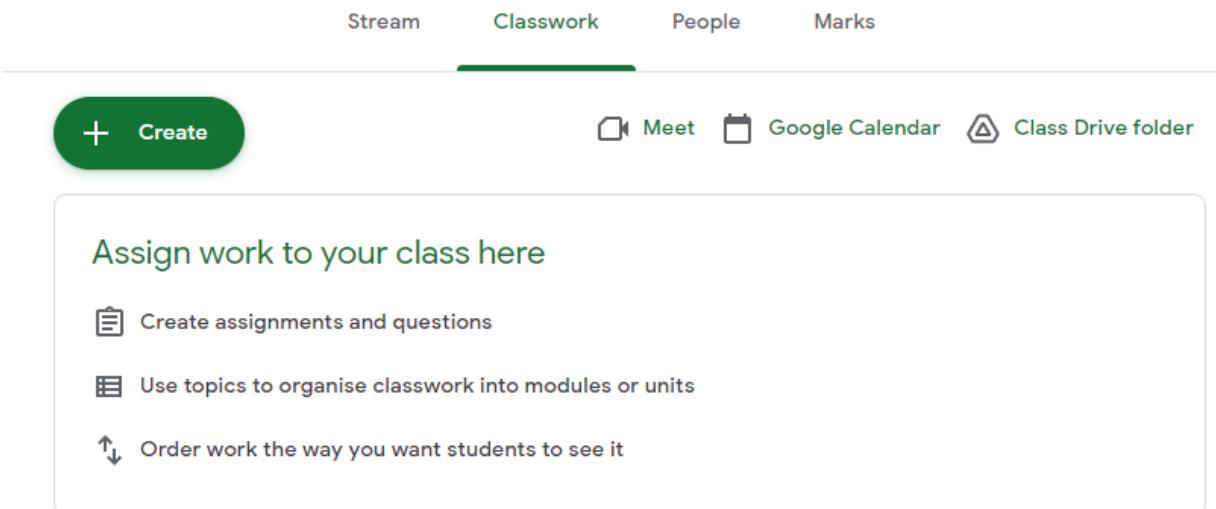


Figure 24 Google Classroom - Class information header

In the classroom view, teachers and students can post for the purpose of sharing documents or teachers post notices to all students in the classroom.



Stream Classwork People Marks

+ Create Meet Google Calendar Class Drive folder

Assign work to your class here

- Create assignments and questions
- Use topics to organise classwork into modules or units
- Order work the way you want students to see it

Figure 25 Google Classroom – Classwork page

This is the interface where the instructor will create the classwork, where all students can submit their documents according to the instructor's requirement

Teachers



(FAID HN-P.TS TopUp HN) Michael Omar



Classmates

132 students



Bach Dang Hoang Quy - FAID HN



Bui Quang Linh - FAID HN



Dang Van Duy - FAID HN



Dao Huu Quyet - FAID HN



Dinh Xuan Hoang - FAID HN



Figure 26 Google Classroom - People page

On the People screen, you can see a complete list of students in the course and who are current instructor.

2.2.3 CMS

CMS is the website that manages the school management website system of Greenwich University Vietnam. Visit website at: <https://cms.greenwich.edu.vn/>

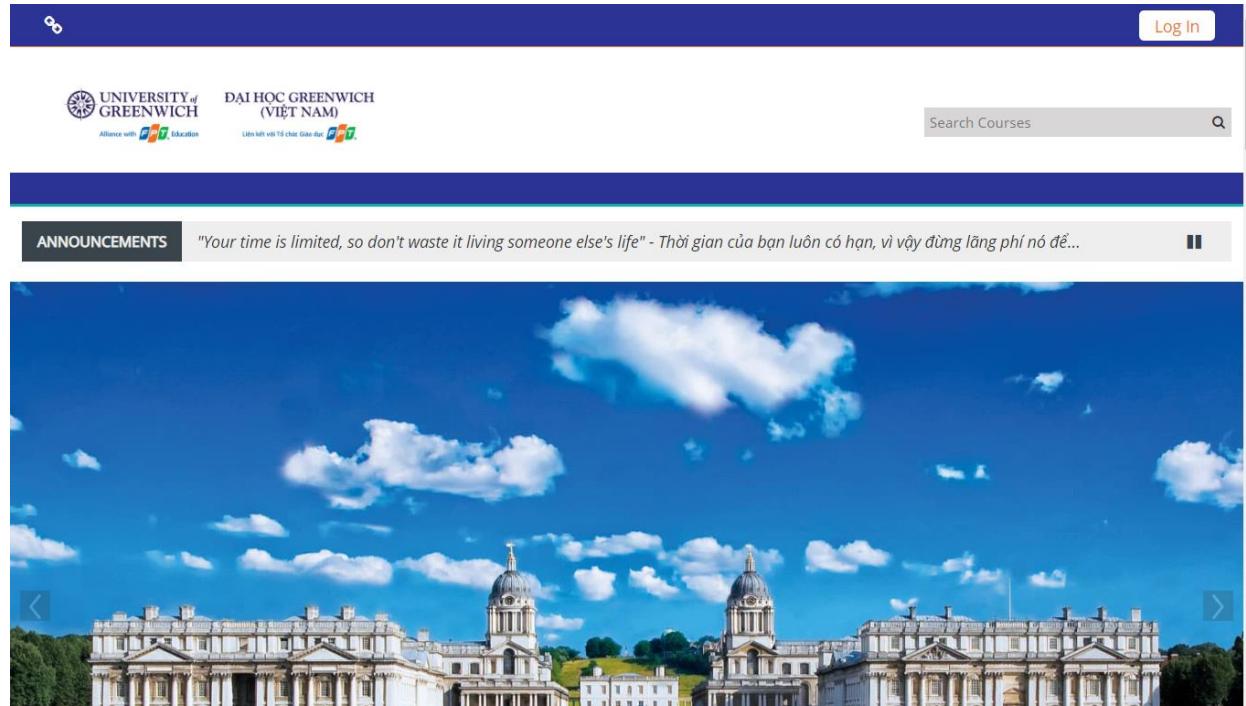


Figure 27 CMS - Home page

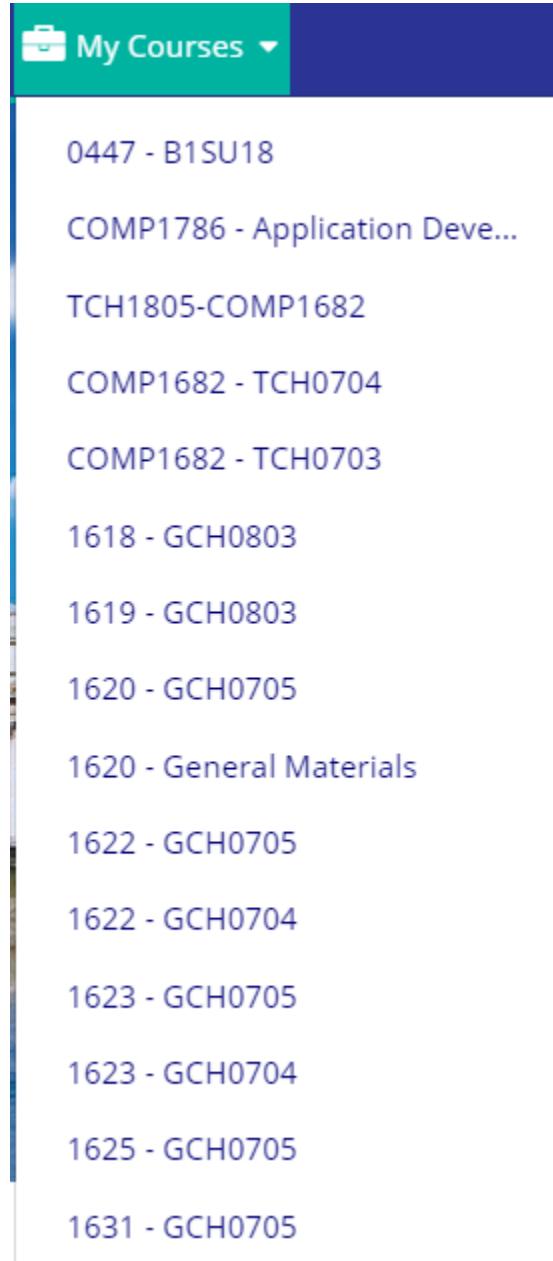
Course categories

[▼ Collapse all](#)

- ▼ Hà Nội (1)
 - ▼ F2G
 - ▶ Computing and Systems Development (1)
 - ▶ [Common Courses](#)
 - ▶ Graphic and Digital Design
 - ▼ TopUp (1)
 - ▶ COMP1649 - Interaction Design (2)
 - ▶ COMP1640 - Enterprise Web Software Development (1)
 - ▶ COMP1714 - Software Engineering Management (2)
 - ▶ COMP1639 - Database Engineering (1)
 - ▶ COMP1108 - Project (7)
 - ▶ COMP 1648 - Development Frameworks and Methods (1)
 - ▶ COMP1661 - Application Development for Mobile Devices (1)
 - ▶ COMP1682 - Project (11)
 - ▶ COMP1787 - Requirement Managements (1)
- ▼ F2G - RQF
 - ▶ Business Program

Figure 28 CMS - Categories

The website offers specific sciences including major faculties such as Business Program, Computing, Graphic Design. Students can keep track of courses related to the faculty they are attending.



The screenshot shows a user interface for managing courses. At the top, there is a green header bar with a white briefcase icon and the text "My Courses ▾". Below this is a dark blue header bar. The main content area is white and lists various course entries, each consisting of a course code and name:

- 0447 - B1SU18
- COMP1786 - Application Deve...
- TCH1805-COMP1682
- COMP1682 - TCH0704
- COMP1682 - TCH0703
- 1618 - GCH0803
- 1619 - GCH0803
- 1620 - GCH0705
- 1620 - General Materials
- 1622 - GCH0705
- 1622 - GCH0704
- 1623 - GCH0705
- 1623 - GCH0704
- 1625 - GCH0705
- 1631 - GCH0705

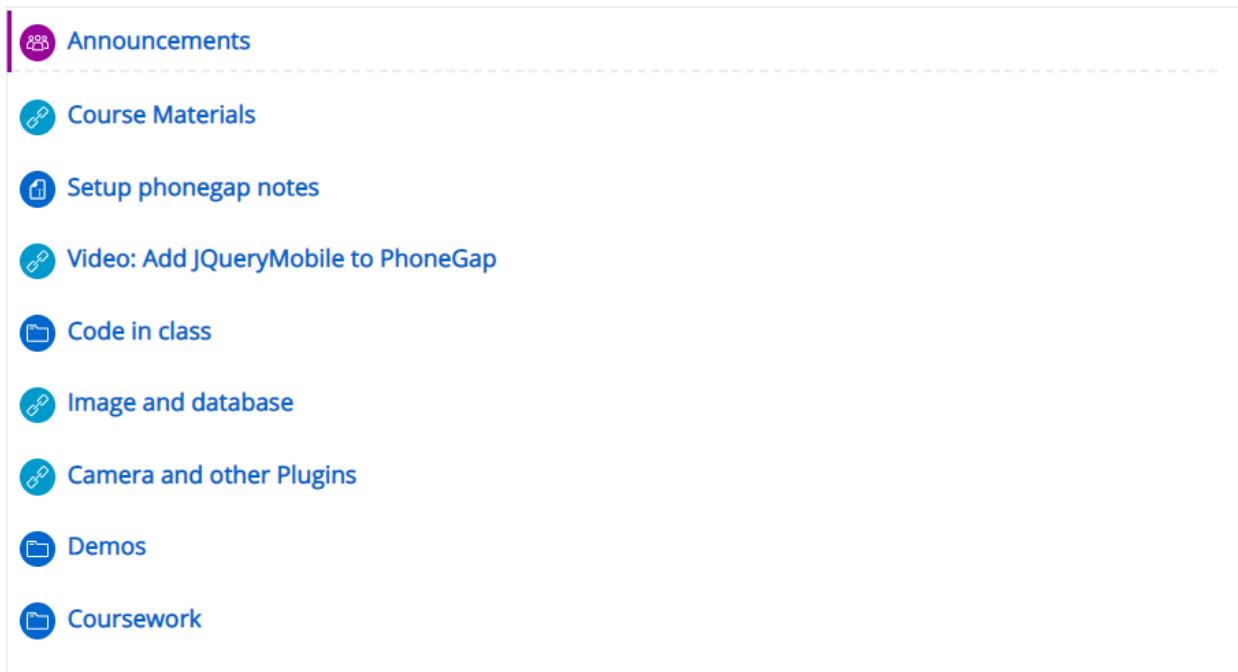
Figure 29 CMS - List own course

In the menu bar, students can keep track of the courses they have been taking



Figure 30 CMS - Join course

When a student joins the class, the student is asked to enter the participation code provided by the instructor at the class.



The screenshot shows a list of course materials on the CMS platform. Each item has a small icon to its left and a blue link to its right. The items are:

- Announcements
- Course Materials
- Setup phonegap notes
- Video: Add JQueryMobile to PhoneGap
- Code in class
- Image and database
- Camera and other Plugins
- Demos
- Coursework

Figure 31 CMS - Activity/Share file in course

In each class, the instructor can create activities that share the document with all students in the class

Submit your report in PDF (only PDF is graded)

Submission status	Submitted for grading
Grading status	Graded
Due date	Tuesday, 3 July 2018, 11:55 PM
Time remaining	Assignment was submitted 3 days 6 hours early
Last modified	Saturday, 30 June 2018, 5:25 PM
File submissions	 Assignment 1 Programming Edited.pdf 30 June 2018, 5:25 PM
Submission comments	 Comments (0)
	Edit submission Remove submission

Figure 32 CMS - Activity Detail

In each activity created by instructors, students can upload their assignment/coursework to each activity.

Grade	Pass
Graded on	Friday, 20 July 2018, 12:11 AM
Graded by	 (Khoa quốc tế) Doãn Trung Tùng
Feedback comments	 Report has a good structure but some contents are not well organized. Paragraphs are not properly aligned. Figures should have captions. ...
Annotate PDF	 Nguyen Pham Thai Son (FGW HN)_17645_0.pdf 20 July 2018, 12:11 AM View annotated PDF...

Figure 33 CMS - Detail grade of activity

Students can also see the results of individual posts they have uploaded. In addition to the functions of learning, managing classes, activities, and scoring system, students can also interact with each other through the messaging system directly on the CMS page.

 **Online users** 

17 online users (last 5 minutes)

-  Kim Quang Nguyễn 
-  Nguyen Van Hoang (FGW DN) 
-  (FGW HCM) Trịnh Ngọc Phúc 
-  Nguyen Duc Anh (FGW HN) 
-  Vo Thi Minh Thu (FGW DN) 
-  Nguyen Ngo Nhu Thao (FGW DN) 
-  Nguyen Pham Thai Son (FGW HN) 
-  Vu Ngoc Loc Tu (FGW HN) 
-  Nguyen Duy Long (FGW HN) 
-  Le Chau Tuan Duc (FGW DN) 
-  Nguyen Thi Quynh Anh (FGW HN) 

Figure 34 CMS - Online student & chat

2.3 Conclusion

Table 2 Comparision of Education websites

Website	Course Videos	Upload blog	Manage class	Send message	Upload activity	Free
Udemy	V	X	X	X	X	V & X
Google Classroom	X	V	V	X	V	V
CMS	X	X	V	V	V	V

Currently, the website about the education system in the country and abroad has appeared a lot on the Internet. However, the website systems mentioned above still lack some functions such as the Udemy website which only allows users to participate in online courses and study at

the website. Google Classroom allows creating classes and helping teachers manage their classes more easily and conveniently. In addition, Google Classroom allows students and teachers in the classroom to post information sharing or share documents to all users in the class so that teachers and students can exchange and share information together. For a CMS system with additional messaging features, students can upload their personal assignments to the system. From the website mentioned above and the functional comparison table, I want to develop an educational website system with the combination of Udemy, Google Classroom, and CMS, helping users to join the courses completely free, online study and able to exchange information, share documents via the blog, students can upload their personal assignments to the system. This is intended to provide a better user experience when using the services of this project developed. The specific functions and requirements of this project will be detailed and clearer in the next section Requirement Analysis.

3 Requirement Analysis

3.1 Introduction

In this part, the study will clearly present the requirements of the educational website system, helping the reader have a better overview of the system. Usecase diagram will also be provided to help readers have a more intuitive view of the objects and functions that will be in this system.

3.2 Requirement

After analyzing similar systems, the Education System project will base on the features available on those similar systems and add other features or say that the Education System is a combined system with outstanding features of Udemy, Google Classroom, and CMS to improve the service quality of the education system website. The system includes three main modules are described in detail in the form of Usecase Diagram below. The usecase diagrams that follow are based on the features mentioned in section **Appendix C – Education System Functionals**

3.2.1 Course module

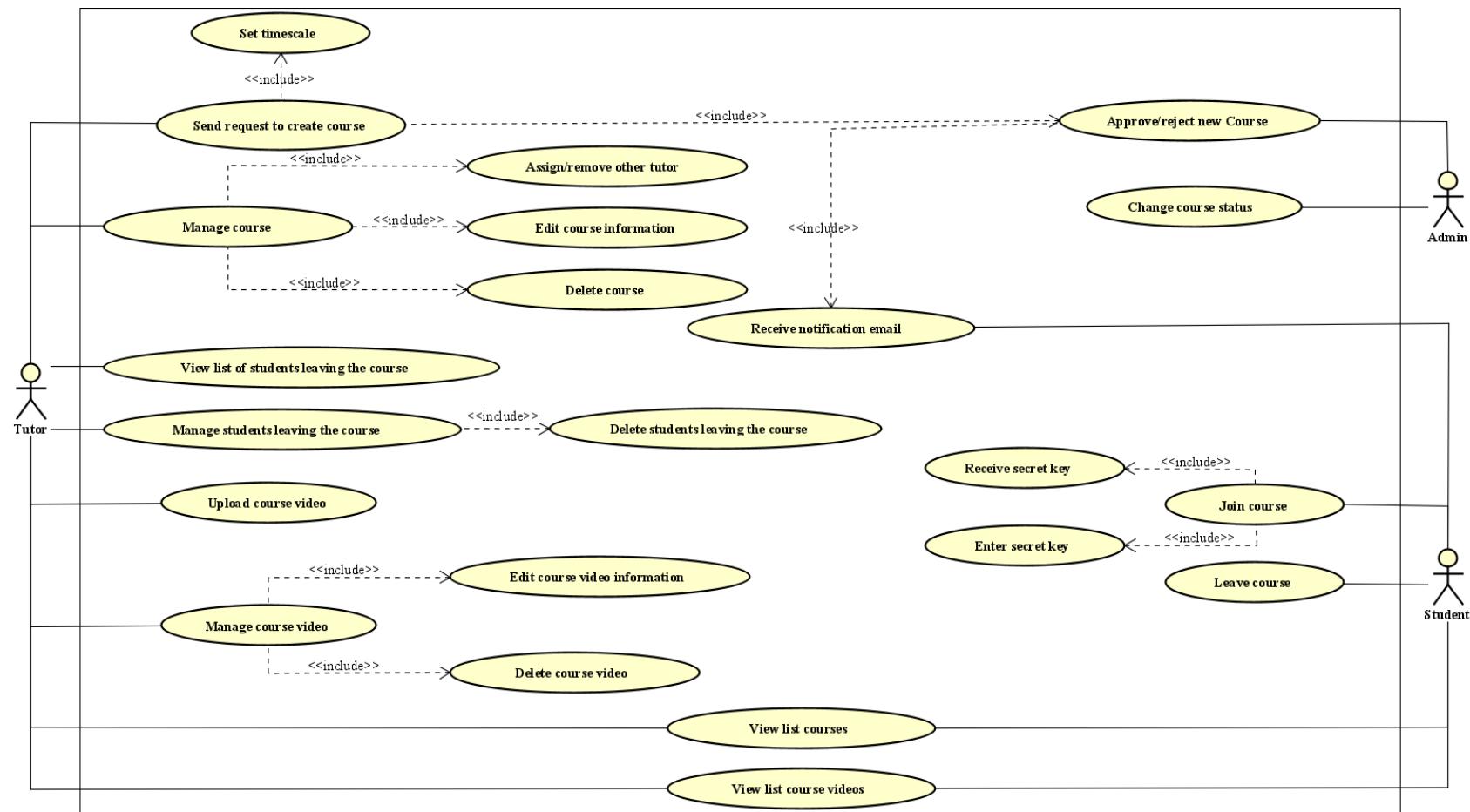


Figure 35 Course module

Since there will be blog modules and activity modules in each course, the course module is the curriculum's main system. As a result, the project will carry out the tasks outlined in the first-course module.

3.2.2 Blog module

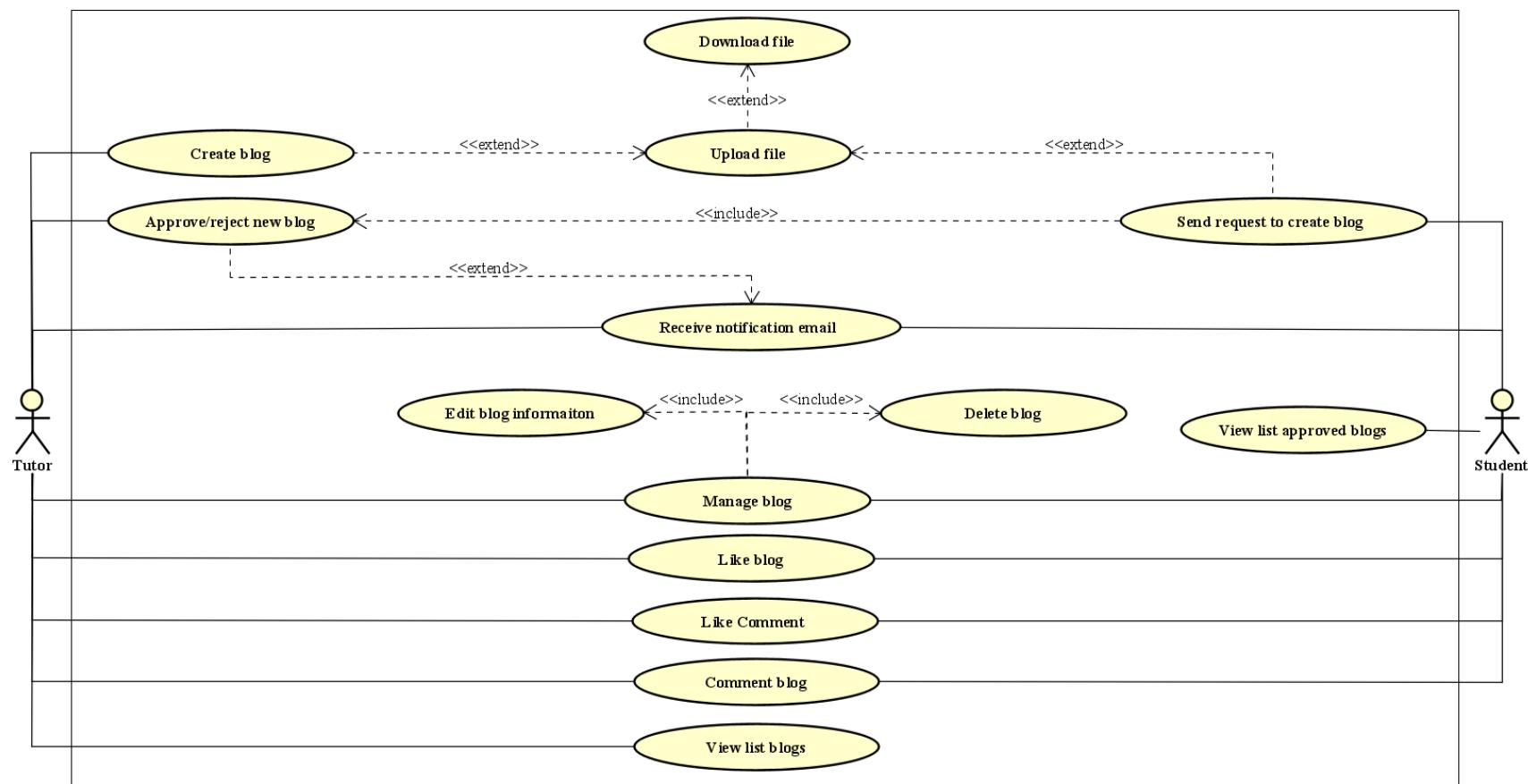


Figure 36 Blog module

After completing the module course, the next system will deploy the blog module. After completing the blog module, the system will deploy the activity module.

3.2.3 Activity module

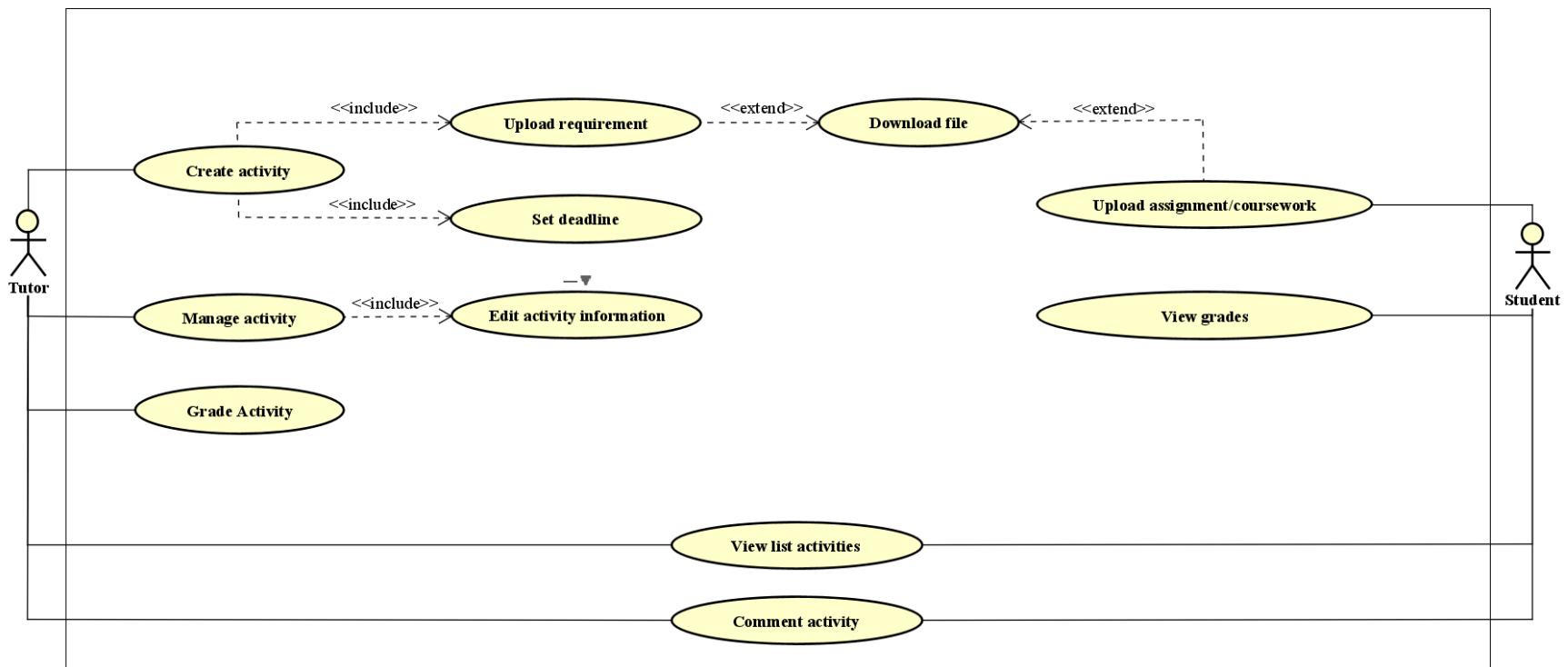


Figure 37 Activity module

These are all main modules of the Education System, below will list in detail the functions of each role in the system.

The Administrator object is the system administrator including the management of users, categories, and courses. The Administrator's detailed functions will be listed below:

- Administrator can login to the system by email and password
- Administrator create the account for the tutor and student
- Administrator can see the list of all the student's account and tutor's account
- Administrator can delete the user account
- Administrator can approve or reject the new course is created by the tutor
- Administrator can change the course's status from On Process to Cancel if this course is reported or receive bad feedback from students
- Administrator can see the statistic report on the dashboard page

The Tutor object is someone who can manage courses, students in the system. Tutor's detailed functions are listed below:

- Tutor can log in to the system by email and password provided by the admin
- Tutor can reset the password via an email link if they forget their password
- Tutor can send the request to create the new course
- Tutor can assign or remove other tutors from course
- Tutor can edit the course information
- Tutor can delete the course
- Tutor can see the list of all the course along with the status:
 - Pending (waiting for administrator's approval)
 - On process (after approving)
 - Accomplish (after the end date of the course)
 - Reject (after teacher send the request to create the course)

- Tutor can see list of all student leave the course with the reason
- Tutor can delete the student from the student list leave the course
- Tutor can upload course videos
- Tutor can see the list of all the course videos
- Tutor can edit course videos information
- Tutor can delete course videos
- Tutor can create a new blog in the course
- Tutor can upload the file with the blog
- Tutor can see the list of all the blog of the student along with the status:
 - Pending
 - Approved
 - Rejected
- Tutor can download file
- Tutor can see list of all own user's blog
- Tutor can approve or reject the new blog of student
- Tutor can edit own blog information
- Tutor can delete own blog and all student's blog
- Tutor can like student's blog
- Tutor can comment on student's blog
- Tutor can create the new activity in the course
- Tutor can see the list of all the activity in the course
- Tutor can see the list of all the assignment/coursework of the student in each activity

- Tutor can edit activity information (edit deadline, limit the number of file upload...) along with the status:
 - On process
 - Overdate
 - Hidden (Can't submit the file by the changing status of tutor)
- Tutor can upload requirement file
- Tutor can receive the notification email whenever student upload
- Tutor can comment on student's activity
- Tutor can grade student's activity
- Tutor can see the list of all the contact
- Tutor can search contact
- Tutor can send message
- Tutor can view statistic report the course is canceled, the course is accomplished, the course is rejected
- Tutor can export PDF the reported statistic

Student object is an object that can participate in courses, upload individual exercises in the system. Student functions will be listed below:

- Student can log in to the system by email and password provided by the admin
- Student can reset the password via an email link if they forget their old password
- Student can receive the notification email when the new course is created
- Student can see the list of all the course along with the status:
 - On process (after approving by admin)
 - Accomplish

- Student can send the request to join the course
- Student can receive the secret key via email
- Student can enter the secret key to the input field to join the course
- Student can leave the course
- Student can receive the notification email when the tutor or other students create a new blog
- Student can upload the file with the blog
- Student can see the list of all the approved blogs in the course
- Student can download file
- Student see list of all own user's blog
- Student can edit own blog information
- Student can delete own blog
- Student can like user's blog
- Student can comment on user's blog
- Student can see the list of all the activity in the course
- Student can upload assignment/coursework file
- Student can see the grade in each activity
- Student can see the list of all the contact
- Student can search contact by name
- Student can send message

3.3 Conclusion

In this section, all functions and features are listed for each user, including the administrator, tutor, and student. All functions listed are intended to provide an overview of the user interaction with the system. All features and functions are built on the analysis of similar systems and the combination of similar systems results in a product that more features for the user experience on website services, the educational system more complete and more convenient. In the next section, the study will show how and process the system design.

4 Software Design

4.1 Introduction

Before going to the website implementation, in this section, we will provide an ERD to help determine the system's database including the properties that will be available for each specific object. This ERD blueprint is based on the requirements listed Requirement Analysis above.

4.2 Conceptual Design

From the requirements analyzed in **Requirement Analysis**, include usecase diagrams that show the main modules of the system and the listed functions. Since the main modules have been shown in the form of usecase diagrams, it is necessary to build tables to store user information, a table to store information of the courses created by the tutor, the blog table to store the blog information in each course, Activity table stores information of activities in each course, category table stores information of categories, in each category contains corresponding courses. In addition, like and comment tables to save information about users performing like and comment functions.

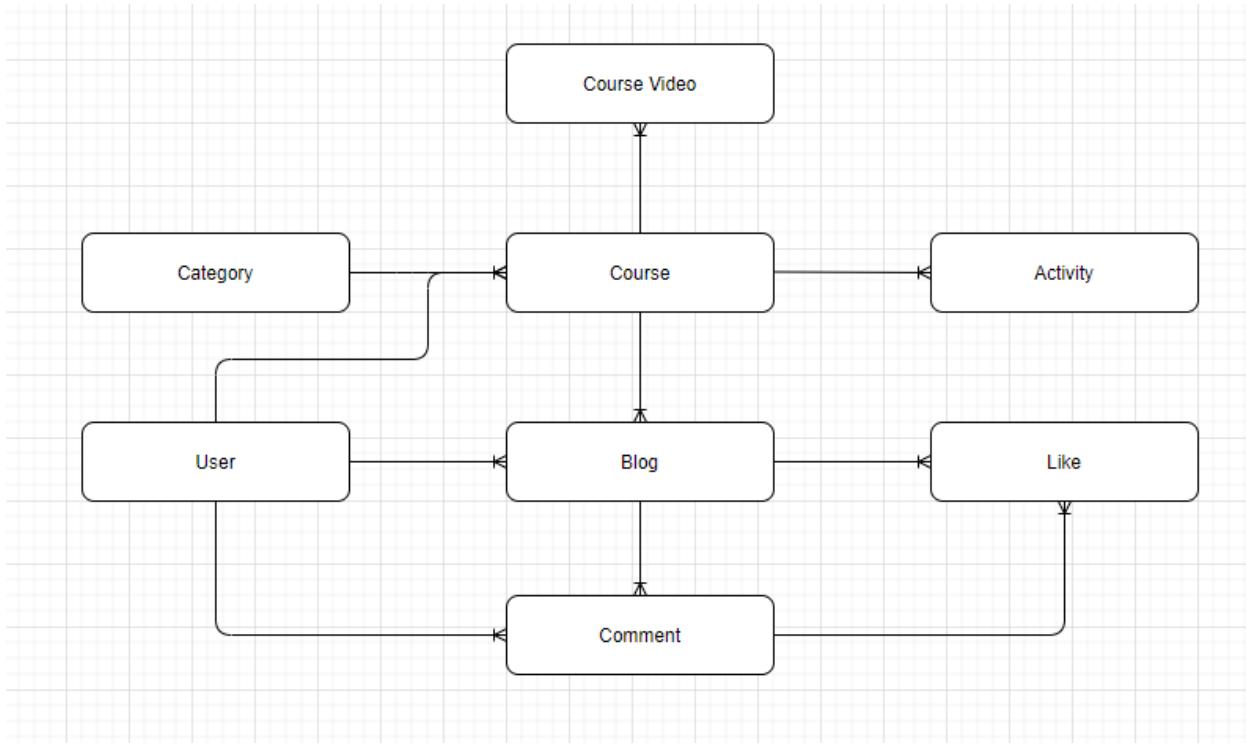


Figure 38 Conceptual Design – Education System

4.3 Entity Relationship Diagram

In this section, the report will provide an explanation of why the following schemas appear. All explanations below are based on the requirement analysis mentioned in the section Requirement Analysis the system functions listed in section **Appendix C – Education System Functionals**

Because the Education System requires users to log into the system to use this system. Therefore, the user table is used to store information of all users in the system. Because the system has 3 main roles including tutor, student, and admin, so the role field is used to distinguish each account belongs to which role, since that account will have a separate display and function for each role. For the account creation function, the admin will be the one who creates accounts for all users of the system and sends emails to users. Users who receive account information and password can log into the system and then the system will ask users to change their password when logging in for the first time. Therefore, the is_active field is used to mark whether the user has changed the password, if the user changes the password the is_active field will be true. In case the user has not changed the password in the first-time login, the is_active field will be true

and each time the user logs in to the system, the website will redirect the user to the password change page. In addition, the fields firstName, lastName, contact, address, email, password store basic information of each user in the system (Figure 39 User schema).

user		...
!	_id	objectId NN
!	firstName	string
!	lastName	string
!	email	string
!	password	string
!	contact	string
!	address	string
!	role	string
!	is_active	bool

Figure 39 User schema

Because each course will belong to a specific category like IT under Programming, there is a need for a category table to store the categories of the system (Figure 40 Category schema).

category		...
!	_id	objectId NN
!	name	string

Figure 40 Category schema

In each category there will be courses related to that category, so a course table is required to store information on the courses.

With the tutor, the function sends a request to create a course and requires the administrator's approval, so the status field to mark what state the course is in, for example, the pending state is the pending admin approval. After the course is approved by the admin, the status will be transferred to the on process. With the status, when the course is canceled or disapproved by the admin when the course is created, the reason field will be updated by the admin.

With the set timescale function, each created course will have a certain time period so fromDate and toDate fields to mark how long that course lasts. After the course finishes the course status will be marked with the status as Accomplish which means the course has been completed.

With the function of participating in a student's course, students need to have a secret key to join the course, so the secretKey field stores the key information of each course. Students can join the course by entering this key from the instructor provided. The categoryId field will be stored in each course to identify which category the course belongs to. In addition, with the assigned other tutor function, this field stores a list of other tutors assigned to that course. Assigned tutors will have the same functionality as the course creator. In addition to createdBy field, categoryId is used to store information about which tutor is created by and in which category (Figure 41 Course schema).

course		...
_id	objectId	NN
createdBy	objectId	NN
categoryId	objectId	NN
name	string	
description	string	
status	string	
fromDate	date	
toDate	date	
secretKey	string	
assigned	array	
reason	string	

Figure 41 Course schema

In each course, the tutor has the function of uploading course videos into each course and managing those video courses, so the video course table is used to store information of the video courses that the tutor has uploaded (Figure 42 Course video schema).

course_video		...
	_id	objectId NN
	courseld	objectId NN
	name	string
	video_file	string

Figure 42 Course video schema

In each course, the system includes a blog posting function for tutors and students. A blog is a place where tutors or students can post blog posts to share knowledge or inform the information of each course. Hence the blog board is the board that stores the information of the blogs in each course. With the function of posting a blog to students need the tutor's approval in the course, so is_approve field is used to mark the student's post status has been approved or not. In addition, the blog post function allows tutors or students to post attachments, so the files field is used to store information on files uploaded by the user. The author and courseld fields to store information about who the blog post is and which course the blog belongs to (Figure 43 Blog schema).

blog		...
	_id	objectId NN
	author	objectId NN
	courseld	objectId NN
	title	string
	content	string
	files	array
	is_approve	bool

Figure 43 Blog schema

In addition, the blog module has other features such as like blogs and like comments, so the like and comment tables are used to store like and comment information (Figure 44 Like & Comment schema).

like		...
■■■ _id	objectId	NN
■■■ blogId	objectId	NN
■■■ commentId	objectId	NN
type	string	

comment		...
■■■ _id	objectId	NN
■■■ author	objectId	NN
■■■ blogId	objectId	NN
content	string	

Figure 44 Like & Comment schema

In each course, the system in addition to functions related to the blog, the system also has functions related to the activity. Activity is where the tutor can create activities and upload associated requirement files and students can upload their personal exercises to each of those activities. Therefore, the activity table is used to store information of the activities in each course.

With the function of creating activities, the tutor can consider the deadline for students to submit, so the fields fromDate and toDate mark the time that the tutor considers for each activity. With the name and description, fields store the name and description information for each activity.

With the function of managing the tutor's activity, the tutor can change the deadline, name, or description, in addition, the tutor can modify the status for each activity to hidden, the status field will have 3 status: on process, out of date, hidden. If the tutor changes the state to hidden, even though within the deadline, students will not be able to upload posts if the tutor changes this state to hidden (Figure 45 Activity schema).

activity		...
■■■ _id	objectId	NN
■■■ courseId	objectId	NN
name	string	
description	string	
files	array	
fromDate	date	
toDate	date	
status	string	

Figure 45 Activity schema

For student's join course function, the user-course table aims to help the system know that the student has joined the course or not. When students access a certain course, the system will check if there exists a student's _id and a course _id or not. In that case, if the record does not exist, the system will email the secret key of that course to the student's Gmail.



Figure 46 User-course schema

From the specific analysis of the above tables, schemas are created to show the relationships of the tables together (Figure 47 EduSys Schemas).

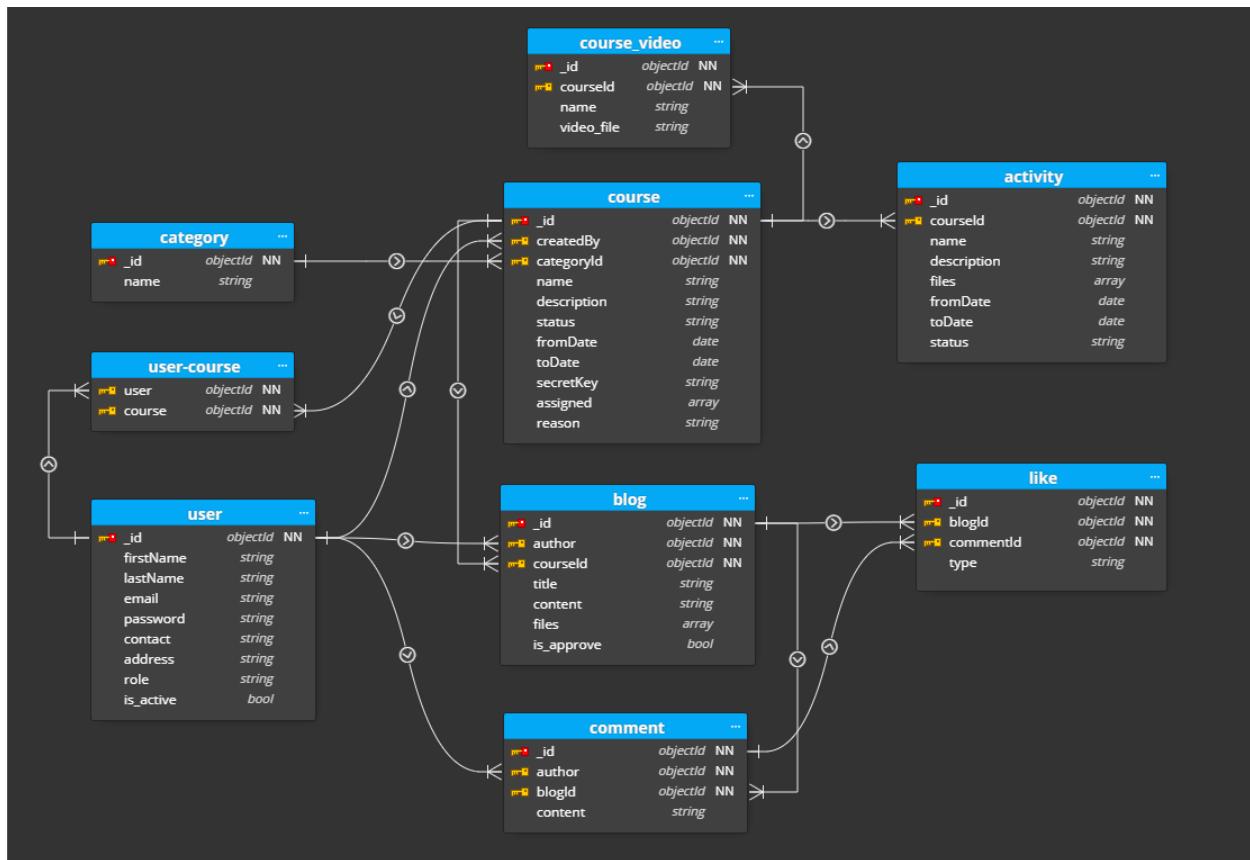


Figure 47 EduSys Schemas

4.4 Activity Diagram

4.4.1 Create new course process

When tutors create a course, the tutors will have to enter enough information to create the course, when the course is created successfully, the new course record will be saved in the database. The admin will then be able to view all the courses created by the tutors. If the admin approves the course, the new course created by the tutors can be displayed on the Education System, then students and tutors can see and join these courses. When the admin has successfully reviewed a new course, all students in the system will receive an email notifying them that a new course has been created.

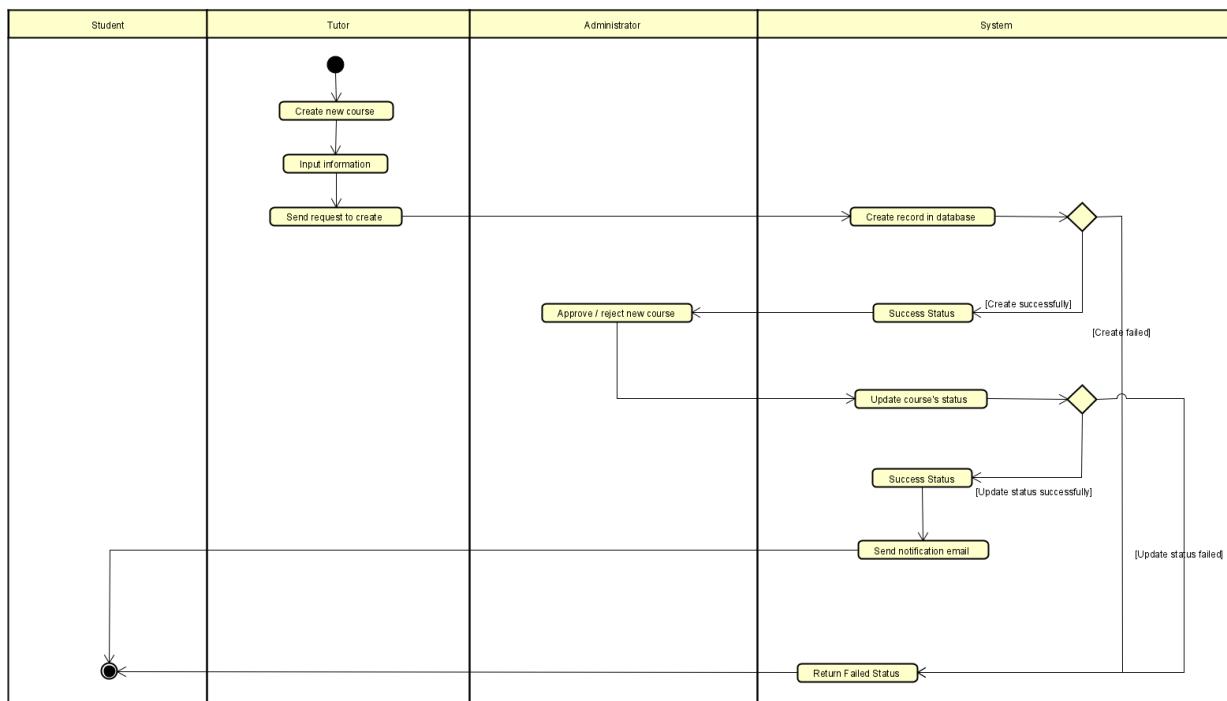


Figure 48 Activity Diagram - Create new course process

4.4.2 Create new blog process

A blog module is a place where tutors or students can post blogs to share information or announcements from tutors in each course. Tutors can post by clicking the new blog button, then enter full information and post. If the poster is a student or student, after entering enough information on the post content and posting, the system will save the record in the database.

Tutors can view all the blogs that students have posted. Tutors have the right to review student blog posts and approve or reject them. When the tutors approve any student blog post, all students in the same course will receive an email notification.

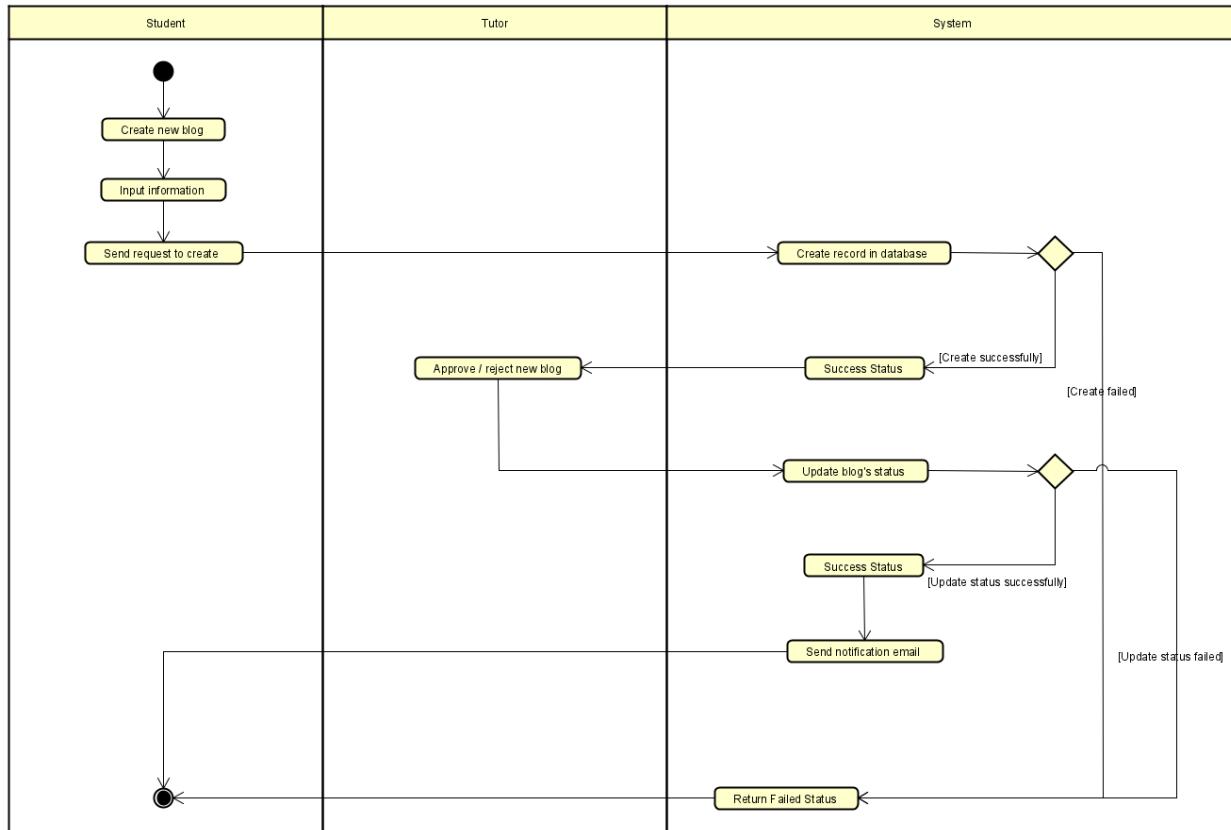


Figure 49 Activity Diagram - Create new blog process

4.4.3 Submit activity file process

When tutors upload a requirement file to activity or students upload file assignment or coursework. The system will search to see if that activity exists, if there exists a file uploaded by tutors or students will be uploaded and stored on s3. After successfully uploading to s3, the file record will be stored in the database.

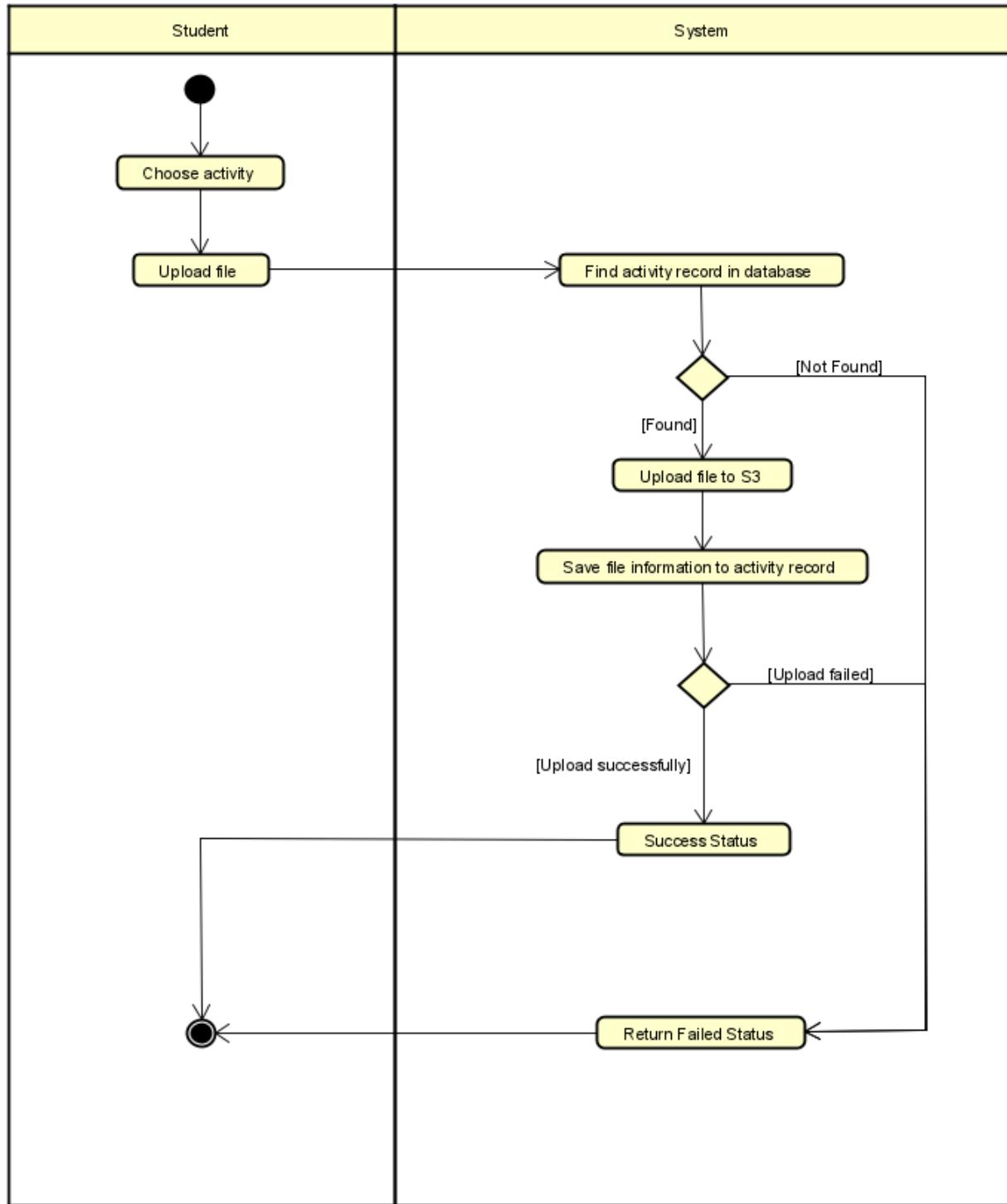


Figure 50 Activity Diagram - Submit activity file process

5 Software implementation

5.1 Introduction

In this section, the implementation of the Education System will be covered including specific steps from starting the project from scratch, installing related packages ... In this section, the report will provide a few examples of how the system wrote code includes a detailed explanation.

5.2 Install Node package management (NPM)

In order to use Javascript language as backend language running on the NodeJs enviroment, developers need to install the Node enviroment. Node package management is a tool to manage and create Javascript programming libraries in a Nodejs environment. NPM allows the installation of libraries or frameworks that have pre-programmed a certain function, helping developers to minimize having to rewrite a certain piece of functionality over and over again. To install a certain library, developers can install the following syntax: "*npm install <package>*".

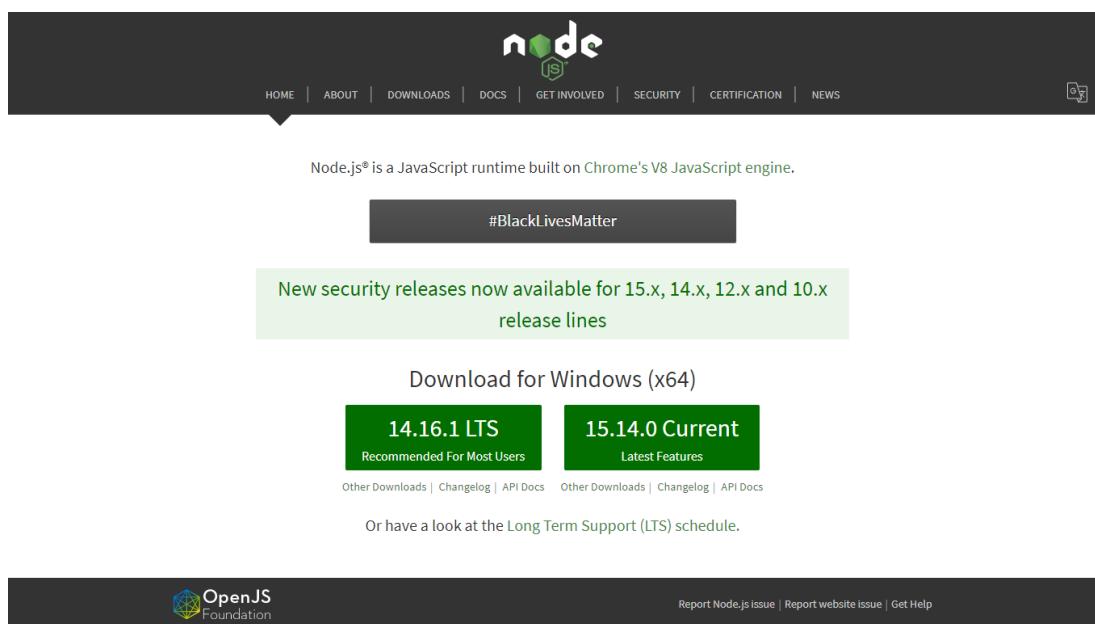
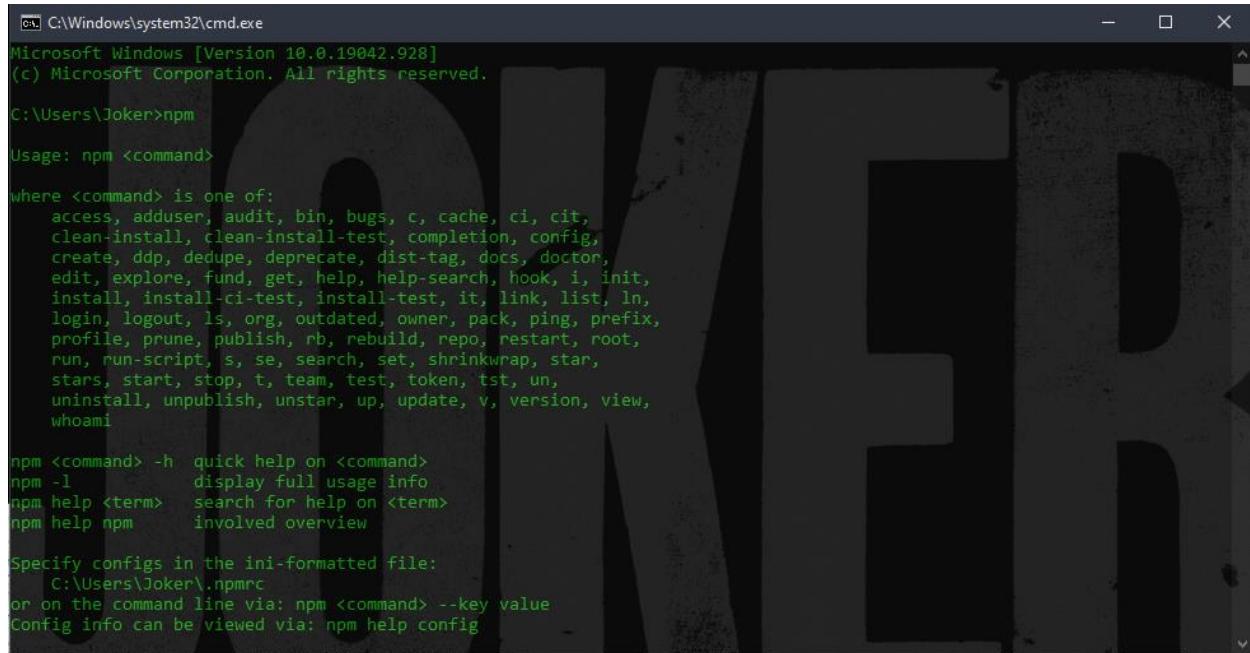


Figure 51 Nodejs Home page

After a successful installation, developers can test with the command in a terminal window with the command `npm`, or "`npm -v`" and "`node -v`". If the result looks like the image below, the environment installation was successful.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

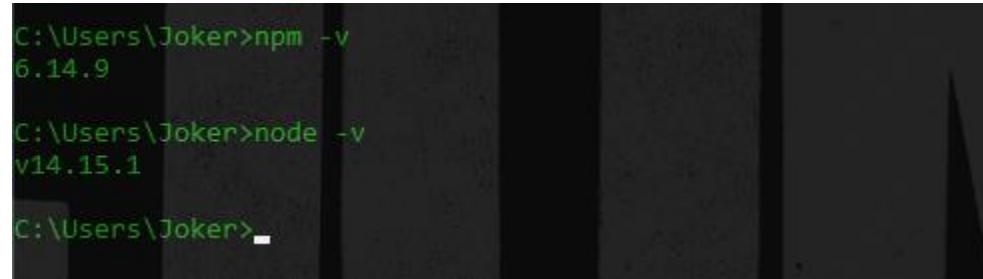
C:\Users\Joker>npm
Usage: npm <Command>

where <Command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
  clean-install, clean-install-test, completion, config,
  create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
  edit, explore, fund, get, help, help-search, hook, i, init,
  install, install-ci-test, install-test, it, link, list, ln,
  login, logout, ls, org, outdated, owner, pack, ping, prefix,
  profile, prune, publish, rb, rebuild, repo, restart, root,
  run, run-script, s, se, search, set, shrinkwrap, star,
  stars, start, stop, t, team, test, token, tst, un,
  uninstall, unpublish, unstar, up, update, v, version, view,
  whoami

npm <command> -h  quick help on <command>
npm -l      display full usage info
npm help <term>  search for help on <term>
npm help npm    involved overview

Specify configs in the ini-formatted file:
  C:\Users\Joker\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config
```

Figure 52 Checking with command npm



```
C:\Users\Joker>npm -v
6.14.9

C:\Users\Joker>node -v
v14.15.1

C:\Users\Joker>
```

Figure 53 Checking with command npm -v and node -v

After successfully installing the environment, developers can install any of the libraries available on the npm library. Developers can search for packages at the npm homepage: <https://www.npmjs.com/>

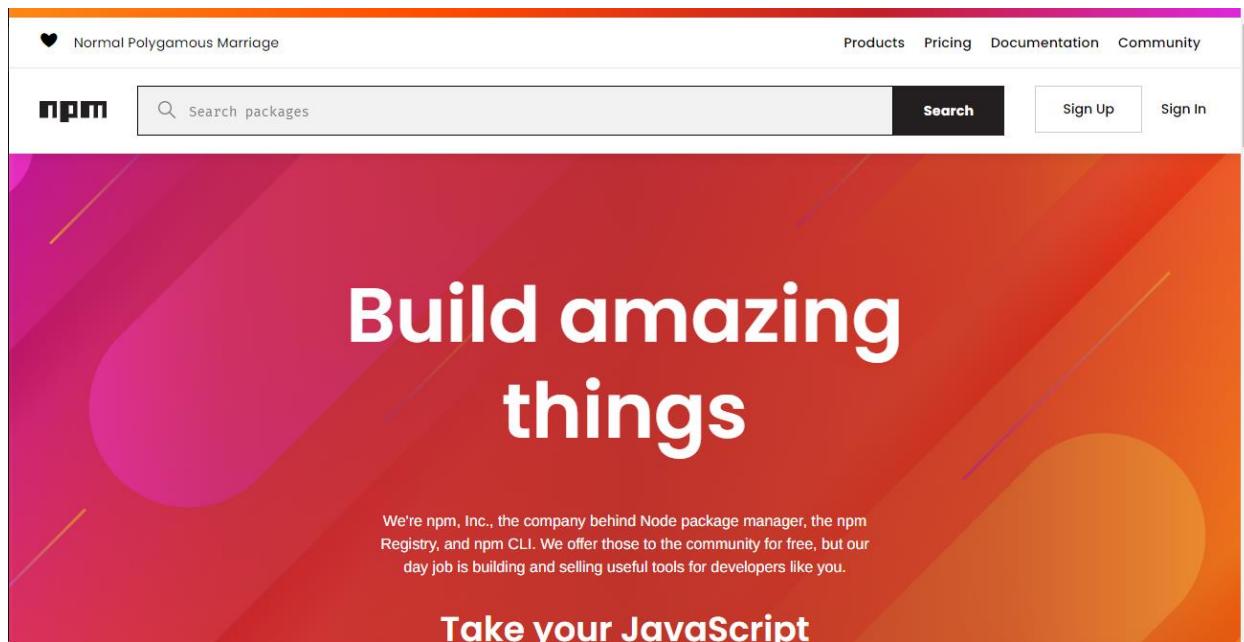


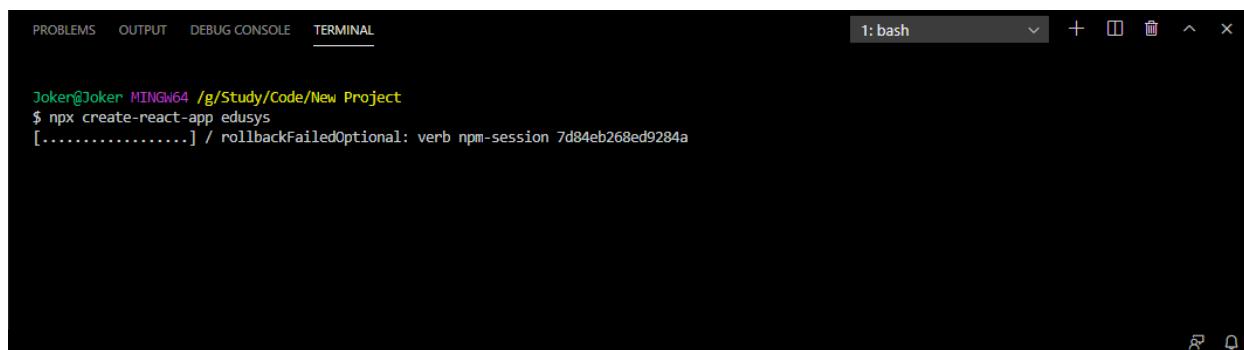
Figure 54 NPM home page

5.3 Install Yarn

Yarn is also a tool that helps developers to install existing libraries, instead of npm having to use the command “*npm install <package>*”, yarn will use the command “*yarn add <package>*”. To be able to install yarn, developers can install it using the command “*npm install yarn -g*”. In that, -g means this set package can be used anywhere.

5.4 Initialize the React app

To initialize the frontend using React, developers will use the command “*npx create-react-app <project-name>*” to be able to create a React project.



The screenshot shows a terminal window with several tabs at the top: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The terminal window has a dark background and displays the following command and output:

```
Joker@Joker MINGW64 /g/Study/Code/New Project
$ npx create-react-app edusys
[.....] / rollbackFailedOptional: verb npm-session 7d84eb268ed9284a
```

Figure 55 Create react app

After successfully launching the React app, the console screen says that the react app was successfully initialized.

```
yarn start
  Starts the development server.

yarn build
  Bundles the app into static files for production.

yarn test
  Starts the test runner.

yarn eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

cd edusys
yarn start

Happy hacking!

© tabnine
```

↙ Prettier ⌂ ⌂

Figure 56 Create react app successful

A successfully instantiated react app will have the following directory structure

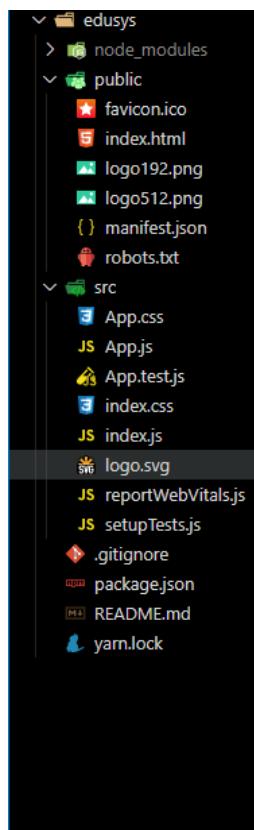


Figure 57 Folder structure of React app

When the React app is successfully launched, the screen will appear as shown Figure 58

Running React app

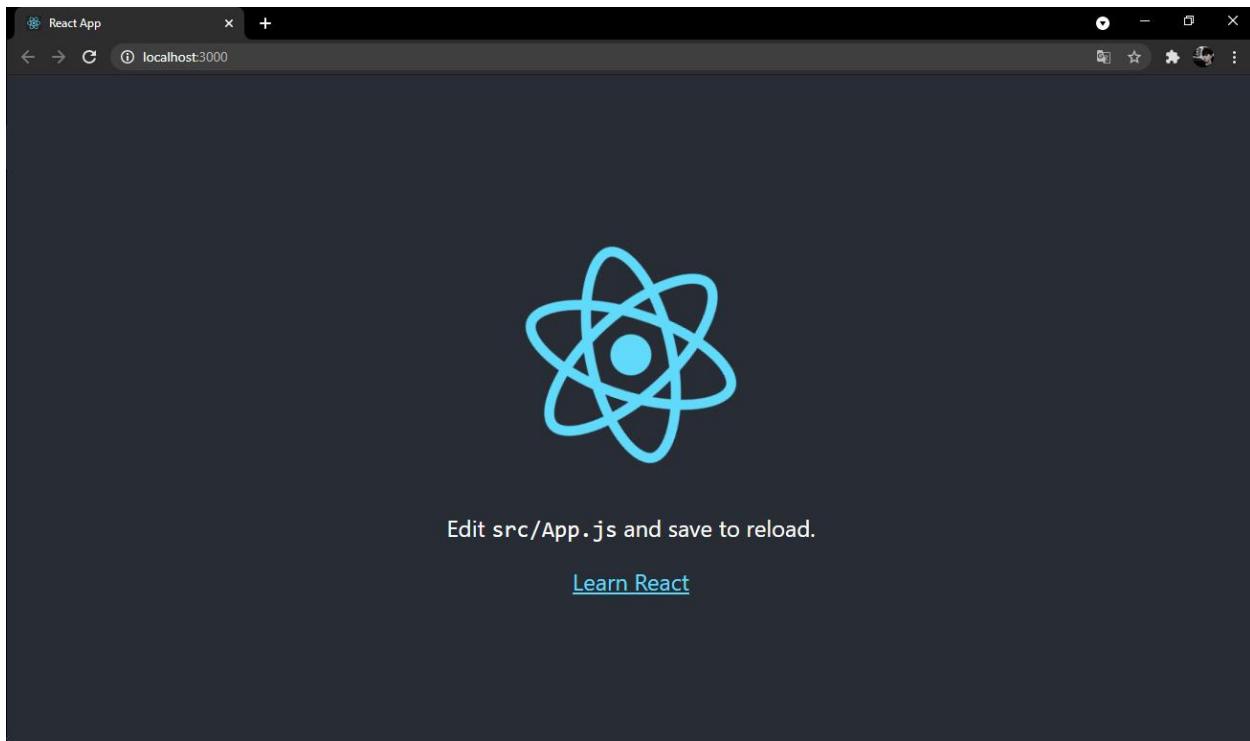


Figure 58 Running React app

Since React handles and works around the state, the large projects will have difficulty handling states, when passing props from parent to child components is complicated. Therefore, the Education System using Redux helps the project to manage states more easily. So, in order to get started with a React project, developers need to set up the necessary folders and files to use Redux.

Create a store, where the state of the system is stored, developers can access the store and get data displayed in the browser or add, edit, and delete them.

```

import { createStore, applyMiddleware } from 'redux';
import thunk from 'redux-thunk';
import { composeWithDevTools } from 'redux-devtools-extension';

import rootReducer from '../reducers';

const userInfoFromStorage = localStorage.getItem('user') ? JSON.parse(localStorage.getItem('user')) : null;

const initialState = {
  auth: {
    user: userInfoFromStorage
  }
};

const middleware = [thunk];

const store = createStore(
  rootReducer,
  initialState,
  composeWithDevTools(applyMiddleware(...middleware))
);

export default store;

```

Figure 59 Redux store

This is an example of creating actions of redux, actions that call API and return data for reducers.

```

import { getToken } from '../utils'
import axios from 'axios';
import { API_CONFIG } from '../config';
import { userConstants } from './constants';
import qs from 'qs';
import imageCompression from 'browser-image-compression';

export const getListUserAction = (filter) => {
}

export const createUserAction = (body) => {
}

export const updateUserAction = ({ id, body }) => {
}

export const updateAvatarAction = ({ id, file }) => {
}

```

Figure 60 Create action of redux

Create constant variables that are used to mark actions to send action so the reducer can capture actions and process data.

```
> export const authConstants = { ...  
  }  
  
> export const userConstants = { ...  
  }  
  
> export const categoryConstants = { ...  
  }  
  
> export const courseConstants = { ...  
  }  
  
> export const blogConstants = { ...  
  }  
  
> export const chatConstants = { ...  
  }  
  
> export const activityConstants = { ...  
  }
```

Figure 61 Constant variables

Create reducers, which capture actions from actions and process data, and return a new state to store.

```
import { userConstants } from '../actions/constants'
// You, 2 weeks ago • add reusable component
const initState = { ... }
export default (state = initState, action) => {
  // eslint-disable-next-line default-case
  switch (action.type) {
    case userConstants.GET_USERS_REQUEST:
      state = { ... }
      break;
    case userConstants.GET_USERS_SUCCESS:
      state = { ... }
      break;
    case userConstants.GET_USERS_FAILURE:
      state = { ... }
      break;
    case userConstants.ADD_USER_REQUEST:
      state = { ... }
      break;
    case userConstants.ADD_USER_SUCCESS:
      state = { ... }
      break;
    case userConstants.ADD_USER_FAILURE:
      state = { ... }
      break;
  }
  return state
}
```

Figure 62 User reducer file

5.5 Initialize backend with express

For server-side initialization, developers can use the command “*npm init*” or “*yarn init*”. Upon successful creation, the file “*package.json*” will be generated.

```
$ yarn init
yarn init v1.22.10
question name (server): EduSys-Server
question version (1.0.0):
question description:
question entry point (index.js): server.js
question repository url:
question author: sonnptgch17274
question license (MIT):
question private:
success Saved package.json
Done in 42.16s.
```

Figure 63 Initialize backend

On the backend side, the folder structure is created with the MVC (models - where schemas are built, the controller - where the requests and responses are processed, and the views are the places where the ejs files are stored for the mailing template).

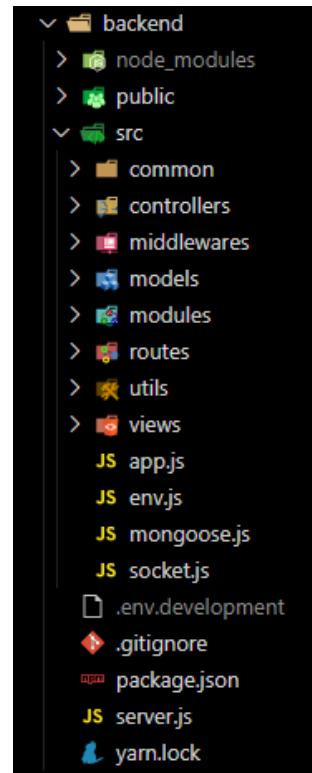


Figure 64 Structure folder of backend



Figure 65 Middlewares

Middlewares is an intermediary handler including authentication - which is the place to confirm the user's access before accessing a certain route.

Validator is a place to check user input when entering data and check the role of each user when accessing a certain route.

Error-handler is where to handle the error returned on the server-side

In the first step, the project uses express, developers can build the server as shown below

```
backend > src > JS app.js > ...
...
1 import './env.js';
2 import './mongoose.js';
3 import './modules/index.js';
4
5 import express from 'express';
6 import morgan from 'morgan';
7 import route from './routes/index.js';
8 import { allowCors, notFound, errorHandler } from './middlewares/index.js';
9
10 import path from 'path'
11
12 const app = express();
13 app.set('port', process.env.PORT);
14
15 app.all('*', allowCors);
16 app.use(morgan('dev'));
17 app.use(express.json());
18 app.use(express.urlencoded({ extended: true }));
19 const __dirname = path.resolve();
20 app.set('views', path.join(__dirname, 'views'))
21
22
23
24 // routing API
25 route(app);
26
27 // error handler
28 app.use(notFound);
29 app.use(errorHandler);
30
31 export default app;
```

Figure 66 Initialize server

Set the environment variables used in the system

```
import dotenv from 'dotenv';
const path = '.env.development';
dotenv.config({ path });
```

Figure 67 Setup use enviroment variables

```
MONGODB_URI="mongodb+srv://edusys:admin@edusys.6sjys.mongodb.net/edusys?retryWrites=true&w=majority"
PORT=5000
AWS_End_Point=s3-ap-southeast-1.amazonaws.com
AWS_ACCESS_KEY=*****
AWS_SECRET_ACCESS_KEY=*****
JWT_SECRET=EduSys-SECRET
SMTP_USER=jokerboy1412@gmail.com
SMTP_PASS=*****
EMAIL_FROM=Education System
SMTP_HOST=smtp.gmail.com
SMTP_PORT=465
SMTP_SERVICE=Gmail
SMTP_USER_NO_REPLY=noreply.
FRONT_END_URL=http://localhost:3000/
```

Figure 68 Setup enviroment variable

Setup function to send mail for the system

```
import nodemailer from 'nodemailer';
import { SMTP_CONFIG, MAILER_FROM } from './enum.js';

const transporter = nodemailer.createTransport({
  host: SMTP_CONFIG.HOST,
  port: SMTP_CONFIG.PORT,
  secure: SMTP_CONFIG.SECURE,
  service: SMTP_CONFIG.SERVICE,
  auth: {
    user: SMTP_CONFIG.AUTH.USER,
    pass: SMTP_CONFIG.AUTH.PASS,
  },
});

export const mailer = ({ email, subject, content }) => {
  return new Promise(async (resolve, reject) => {
    try {
      const data = await transporter.sendMail({
        from: MAILER_FROM,
        to: email,
        subject,
        html: content,
      });
      return resolve(data);
    } catch (error) {
      return reject(error);
    }
  });
};
```

You, 3 weeks ago • set up the backend project

Figure 69 Setup malier function

Here is an example of how the Education System project constructs the schema for users' information.

```
const UserSchema = new mongoose.Schema({
    email: { type: String, unique: true, required: true },
    password: { type: String, required: true },
    passwordResetToken: {
        type: String,
        default: null
    },
    profile: {
        firstName: { type: String, required: true },
        lastName: { type: String, required: true },
        role: {
            type: String,
            enum: ['admin', 'student', 'tutors'],
            default: 'student',
        },
        city: String,
        address: String,
        phone: String,
        avatar: {
            type: String,
            default: 'default_avatar.jpg',
        },
    },
    isDeleted: { type: Boolean, default: false }
}, { timestamps: true });
```

Figure 70 Creating schemas for users

5.6 Create the mongodb

To be able to use MongoDB, developers need to create a project at the MongoDB home page. First, developers must initialize the MongoDB project as shown Figure 71 Create the project mongodb.

Create a Project

Name Your Project > Add Members

Next

Name Your Project

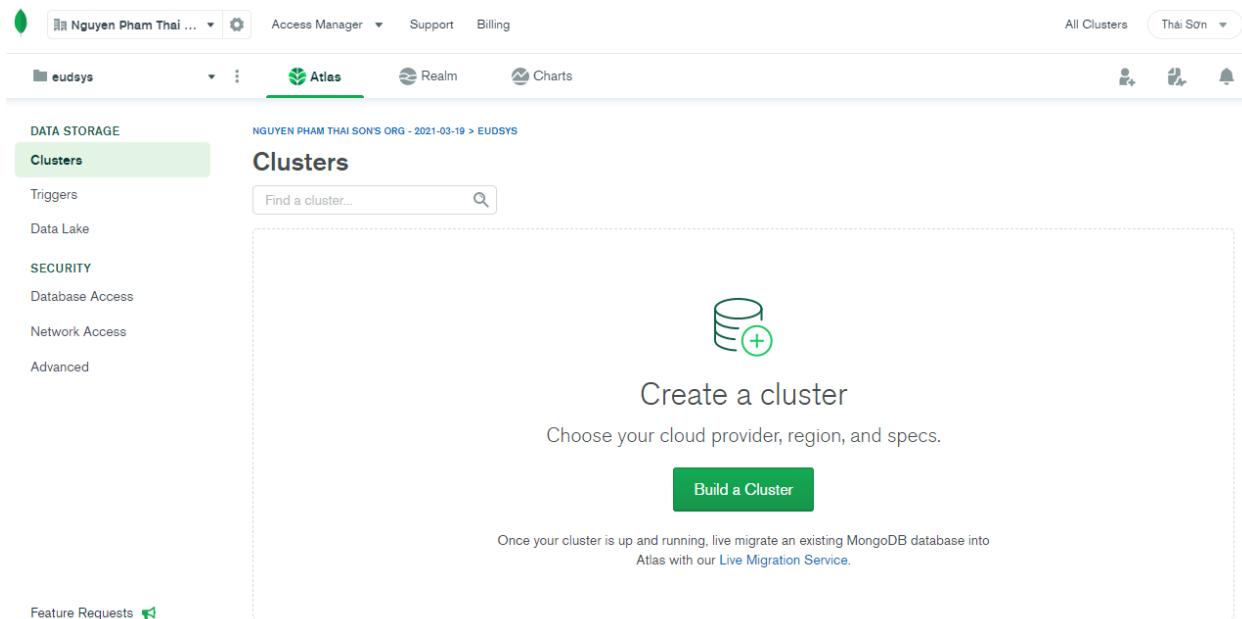
Project names have to be unique within the organization (and other restrictions).

eudsys

Cancel **Next**

Figure 71 Create the project mongodb

After successfully creating the project, the screen will display as shown below Figure 72 Create a cluster.



The screenshot shows the MongoDB Atlas interface. At the top, there's a navigation bar with 'Nguyen Pham Thai ...', 'Access Manager', 'Support', 'Billing', 'All Clusters', and 'Thai Son'. Below the navigation is a header with 'eudsys' and tabs for 'Atlas', 'Realm', and 'Charts'. On the left, a sidebar has sections for 'DATA STORAGE' (with 'Clusters' selected), 'Triggers', 'Data Lake', and 'SECURITY' (with 'Database Access', 'Network Access', and 'Advanced'). The main area is titled 'Clusters' and shows a search bar with 'Find a cluster...'. In the center, there's a green circular icon with a plus sign and the text 'Create a cluster'. Below it, a sub-instruction says 'Choose your cloud provider, region, and specs.' with a 'Build a Cluster' button. At the bottom, a note says 'Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#)'. A 'Feature Requests' link is at the very bottom.

Figure 72 Create a cluster

Next, developers need to create a cluster so that they can store system information in the form of objects or schemas.





★ Recommended region [i](#)

ASIA	EUROPE	NORTH AMERICA
 Mumbai (ap-south-1)	 Ireland (eu-west-1) ★	 Oregon (us-west-2) ★
 Singapore (ap-southeast-1) ★	 Frankfurt (eu-central-1) ★	 N. Virginia (us-east-1) ★
AUSTRALIA		
 Sydney (ap-southeast-2) ★		

Cluster Tier
M0 Sandbox (Shared RAM, 512 MB Storage) >

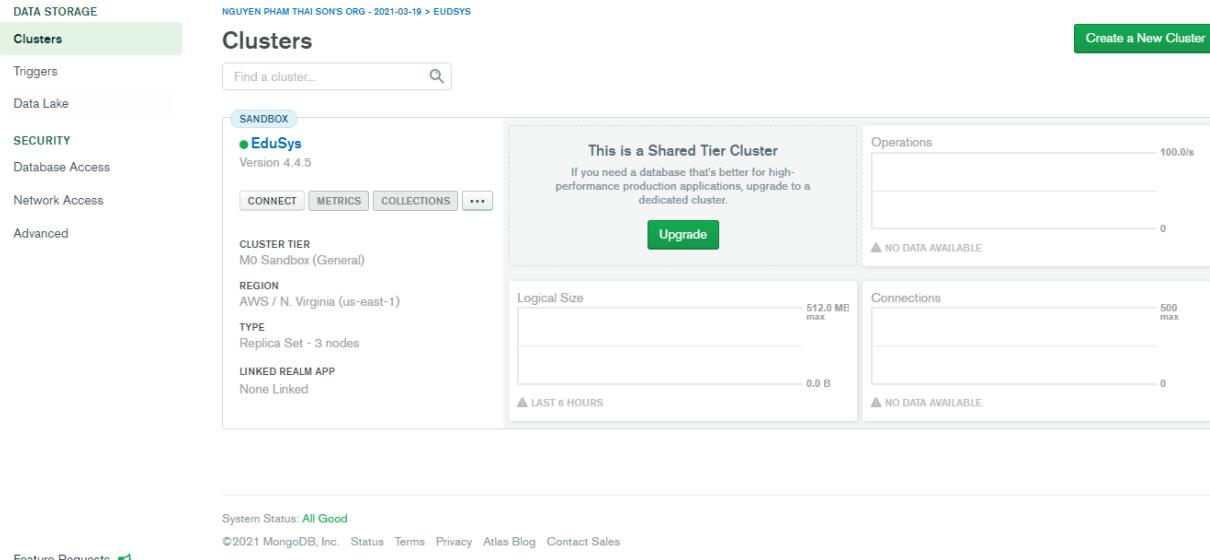
Encrypted

Additional Settings
MongoDB 4.4, No Backup >

Cluster Name
EduSys ▾

FREE
Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.
[Back](#)
Create Cluster

After successfully creating mongodb, the screen will display as below



The screenshot shows the MongoDB Atlas interface. On the left, a sidebar lists 'DATA STORAGE' sections: 'Clusters' (selected), 'Triggers', 'Data Lake', 'SECURITY', 'Database Access', 'Network Access', and 'Advanced'. The main content area is titled 'Clusters' and shows a single cluster named 'EduSys' (Version 4.4.5) in a 'SANDBOX' tier. A callout box highlights that it's a 'Shared Tier Cluster' and suggests upgrading to a dedicated cluster. Below this, there are sections for 'Logical Size' (512.0 MB max, 0.0 B last 6 hours), 'Operations' (100.0/s), 'Connections' (500 max, 0 B last 6 hours), and a search bar. At the bottom, there are footer links for 'System Status: All Good', 'Feature Requests', and '©2021 MongoDB, Inc.'.

Figure 73 Create mongodb successful

5.7 Install package

After creating the project on the client and server-side, developers can install a number of packages to serve the project development.

```
"dependencies": {
    "aos": "^2.3.4",
    "axios": "^0.21.1",
    "browser-image-compression": "^1.0.14",
    "cra-template": "1.1.2",
    "moment": "^2.29.1",
    "moment-timezone": "^0.5.33",
    "qs": "^6.10.1",
    "query-string": "7.0.0",
    "react": "^17.0.1",
    "react-bootstrap": "1.5.2",
    "react-dom": "17.0.1",
    "react-file-icon": "1.0.0",
    "react-icons": "4.2.0",
    "react-paginate": "7.1.2",
    "react-player": "2.9.0",
    "react-redux": "7.2.2",
    "react-router-dom": "5.2.0",
    "react-scripts": "4.0.3",
    "react-spinners": "0.10.6",
    "redux": "4.0.5",
    "redux-devtools-extension": "2.13.9",
    "redux-thunk": "2.3.0",
    "socket.io-client": "4.0.1",
    "sweetalert": "2.1.2",
    "web-vitals": "1.1.1"
},
```

Figure 74 Front-end - Install packages

```
"dependencies": {
  "adm-zip": "^0.5.4",
  "aws-sdk": "^2.863.0",
  "bcrypt": "^5.0.1",
  "bcryptjs": "^2.4.3",
  "body-parser": "^1.19.0",
  "cors": "^2.8.5",
  "dotenv": "^8.2.0",
  "ejs": "^3.1.6",
  "express": "^4.17.1",
  "express-validator": "^6.10.0",
  "jsonwebtoken": "^8.5.1",
  "md5": "^2.3.0",
  "mongodb": "^3.6.6",
  "mongoose": "^5.12.0",
  "morgan": "^1.10.0",
  "multer": "^1.4.2",
  "multer-s3": "^2.9.0",
  "node-schedule": "^2.0.0",
  "nodemailer": "^6.5.0",
  "nodemon": "^2.0.7",
  "socket.io": "^4.0.1",
  "uuid": "^8.3.2"
}
```

Figure 75 Backend - Install package

5.8 Set up upload and download files from s3

In order to be able to interact with a third party, namely the s3 backup service provided by Amazon Web Services, developers need to install the package using the following command: “*yarn add aws-sdk*” or “*npm install aws-sdk*” (sdk means Software Development Kit is a tool that makes developers easier to interact with a platform's third-party service through writing code).

After successfully installing the package, developers need to declare three variables related to s3 to be able to connect to the s3 bucket including: endpoint, access key, and secret access key.

```
export const AWS_ACCESS_BUCKET = {
  endpoint: process.env.AWS_END_POINT,
  accessKeyId: process.env.AWS_ACCESS_KEY,
  secretAccessKey: process.env.AWS_SECRET_ACCESS_KEY,
};
```

Figure 76 AWS_ACCESS_BUCKET

After setting the relevant variables, to be able to upload files to the s3 bucket, developers need to provide the bucket, key, and buffer names. For the download function, developers need to provide a bucket, and key to be able to download files from s3.

```
import AWS from 'aws-sdk';
import { AWS_ACCESS_BUCKET } from './enum.js';

const s3 = new AWS.S3(AWS_ACCESS_BUCKET);

export const uploadAWS = (bucket, key, buffer) => {
    return new Promise((resolve, reject) => {
        s3.upload({
            client: s3,
            Bucket: bucket,
            Key: key,
            Body: buffer,
        }, (err, data) => {
            if (err) {
                reject(err);
            }
            resolve(data);
        });
    });
};

export const downloadAWS = (bucket, key) => {
    return new Promise((resolve, reject) => {
        s3.getObject({
            Bucket: bucket,
            Key: key,
        }, (err, data) => {
            if (err) {
                reject(err);
            }
            resolve(data);
        });
    });
};
```

Figure 77 Uploading and Downloading file from s3 bucket

5.9 Example of implementing the API

5.9.1 API create course

Course creation is one of the main functions of the system, below describes how the API is implemented.

```

export const createCourseService = async (data, currentUser) => {
  const response = {
    statusCode: 201,
    message: 'Create course successful',
    data: {}
  }
  try {
    // check exist category
    const category = await Category.findOne({
      _id: data.category,
      isDeleted: false
    })

    if (!category) {
      return {
        statusCode: 404,
        message: 'Category not found',
        data: {}
      }
    }

    // check exist course
    const checkCourse = await Course.findOne({ title: data.title, isDeleted: false })
    if (checkCourse) {
      return {
        statusCode: 400,
        message: 'Course existed',
        data: {}
      }
    }

    const course = await Course.create({
      title: data.title,
      description: data.description,
      fromDate: data.fromDate,
      toDate: data.toDate,
      category: data.category,
      createdBy: currentUser._id,
      secretKey: uuidv4()
    })

    response.data = await course
      .populate({ path: 'createdBy', select: 'email profile.firstName profile.lastName profile.avatar' })
      .populate({ path: 'category', select: 'name' })
      .execPopulate()

  } catch (err) {
    response.statusCode = 500;
    response.message = err.message;
  }
  return response
}

```

Figure 78 API create new course

Initially, when the user sends a request to create a course, the server will check to see if the information such as category or course exists or not. If the information submitted by the user is correct and exists, the system will check to see if the course creation information is complete, if the information is complete, the server will create a new course and return a response with the newly created course information. (Figure 79 Create course success)

```
1  {
2      "statusCode": 201,
3      "message": "Create course successful",
4      "data": {
5          "status": "pending",
6          "bgImage": "course.jpg",
7          "assigned": [],
8          "isDeleted": false,
9          "_id": "607ee4460d0ef5235cc69576",
10         "title": "Typescript from zero to hero",
11         "description": "Typescript from zero to hero",
12         "fromDate": "2021-03-29T17:00:00.000Z",
13         "toDate": "2021-05-29T17:00:00.000Z",
14         "category": {
15             "_id": "606208596c0f4f1fc042d19d",
16             "name": "Programming"
17         },
18         "createdBy": {
19             "profile": {
20                 "avatar": "Doan Trung-Tung.a677bfe4f2634aa1a0cf57f319440a83.blob",
21                 "firstName": "Doan Trung",
22                 "lastName": "Tung"
23             },
24             "_id": "6079cf906b77d314f47589e5",
25             "email": "tungdt@gmail.com"
26         },
27         "secretKey": "fa03d990-8794-4cec-a553-c5d3f375ba39",
28         "createdAt": "2021-04-20T14:25:10.435Z",
29         "updatedAt": "2021-04-20T14:25:10.435Z",
30         "__v": 0
31     }
32 }
```

Figure 79 Create course success

5.9.2 Join course function

The join course function works for students when they want to see the details of a certain course. The server will check if that student already has a record that includes the student's information

and the course that the student wants to see. If no records exist, the server will send a new response saying "You haven't join this course"

```
export const getCourseService = async (courseId, user) => {
  const response = {
    statusCode: 200,
    message: 'Showing course successful',
    data: {},
  };
  try {
    if (user.profile.role === 'student') {
      const checkUser = await UserCourse.findOne({
        course: courseId,
        user: user._id,
      });
      if (!checkUser) {
        return {
          statusCode: 404,
          message: `You haven't join this course`,
          data: {},
        };
      }
    }
  }
}
```

Figure 80 Get course detail API

When a student has not joined the course, the website will automatically call requestToJoinCourse API so that the student can receive the course key via gmail. (Figure 81 Receive secret key of the course through email).

```

export const requestToJoinCourseService = async (userId, courseId) => {
  const response = {
    statusCode: 200,
    message: 'Success',
    data: {}
  }
  try {
    const user = await User.findOne({ _id: userId })
    const course = await Course.findOne({ _id: courseId })
    if (!user) {
      return {
        statusCode: 404,
        message: 'User not found',
        data: {}
      }
    }
    if (!course) {
      return {
        statusCode: 404,
        message: 'Course not found',
        data: {}
      }
    }
    const body = await renderFile('src/views/join-course.template.ejs', {
      firstName: user.profile.firstName,
      courseName: course.title,
      secretKey: course.secretKey
    })

    await mailer({
      email: user.email,
      subject: 'Request to join course',
      content: body,
    });

    } catch (err) {
      response.statusCode = 500;
      response.message = err.message;
    }
  return response
}
  
```

Figure 81 Receive secret key of the course through email

After verifying that the user and course exist, the server will render the email ejs template file (Figure 82 Email template) already built including the parameters firstName, courseName, secretKey. Where firstName is the name of the student who wants to take part in the course, courseName is the name of the course that the student is requesting to join. In each course, there will be a secret key, students who want to participate in a certain course will have to enter the

exact secret key of that course. Then the server will use the Gmail sending service to be able to email the information of the course secret key that the student is trying to join via Gmail of that student.

```

backend > src > views > <% join-course.template.ejs > ...
34   </head>
35
36   <body>
37     <table align="center" class=MsoTableGrid border=0 width="600" cellspacing="0" cellpadding="0" style='width: 450.0pt; border-collapse: collapse; border: none'>
38       <tr>
39           <td width="600" colspan=2 valign=top style='width: 450.0pt; border: none; padding: 0in 5.4pt 0in 5.4pt; height: 169.4pt'>
40               <p class=MsoNormal style="margin-bottom:0in; line-height: normal">
41                   
42               </p>
43           </td>
44       </tr>
45   </table>
46   <p><br></p>
47   <table align="center" class=MsoTableGrid border=0 width="500" cellspacing="0" cellpadding="0" style='width: 450.0pt; border-collapse: collapse; border: none'>
48       <tr>
49           <td>
50               <table cellspacing="0" cellpadding="0">
51                   <tr>
52                       <td>
53                           <p>Hi <strong>
54                               <%=firstName%>,
55                               </strong></p>
56                           <p>You recently requested to join <strong><%=courseName%></strong> course, please enter the secret key below to join course</p>
57                           <p>Secret key: <strong><%=secretKey%></strong></p>
58                       </td>
59                   </tr>
60               </table>
61           </td>
62       </tr>
63   </table>
64   <p><br></p>
65   <table align="center" class=MsoTableGrid border=0 width="500" cellspacing="0" cellpadding="0" style='width: 450.0pt; border-collapse: collapse; border: none'>
66       <tr>
67           <td>
68               <table cellspacing="0" cellpadding="0">
69                   <tr>
70                       <td>
71                           <p>Sincerely,</p>
72                           <p>Education System Team</p>
73                       </td>
74                   </tr>
75               </table>
76           </td>
77       </tr>
78   </table>
79 </body>
80 </html>

```

Figure 82 Email template

The email sent will be displayed as shown below



Figure 83 Send email service

Students wishing to participate in the course will have to enter the secret key sent by email. When the student finishes entering the secret key and sending the request, the server will check to see if the submitted information such as user information, course information exists or not and that secret key matches the secret key of the course that the student wants to join or not. When the submitted information is correct, the server will create a record that includes the student's information and the course information to confirm that the student took the course.

```

export const joinCourseService = async (userId, courseId, secretKey) => {
    const response = {
        statusCode: 200,
        message: 'Join course successful',
        data: {}
    }
    try {
        const user = await User.findOne({ _id: userId })
        const course = await Course.findOne({ _id: courseId })
        if (!user) {
            return {
                statusCode: 404,
                message: 'User not found',
                data: {}
            }
        }
        if (!course) {
            return {
                statusCode: 404,
                message: 'Course not found',
                data: {}
            }
        }

        const checkUserOnCourse = await UserCourse.findOne({ user: userId, course: courseId })
        if (checkUserOnCourse) {
            return {
                statusCode: 400,
                message: 'Already joined course',
                data: {}
            }
        }

        if(secretKey !== course.secretKey){
            return {
                statusCode: 400,
                message: 'Wrong secret key',
                data: {}
            }
        }

        const joinCourse = await UserCourse.create({
            user: userId,
            course: courseId
        })
        response.data = joinCourse
    } catch (err) {
        response.statusCode = 500;
        response.message = err.message;
    }
    return response
}

```

Figure 84 Join course API

5.9.3 Update the status for course and status when the deadline is over

The system uses node-schedule to automatically run a certain function every 0:00 every day. In the system, the course module and activity module are used node-schedule to have the status change automatically every time the toDate is smaller than the current date. For the course module, if the toDate is less than the current date, the course status will be changed from the on process status to the accomplish status, which means the course is complete. For the activity module, when to date is smaller than the current date, activity status will be changed from on

process status to outDate status, which means that activity has expired, students cannot submit assignments/coursework.

The ON_EVERY_DAY variable represents 0 seconds, 0 minutes, 0 hours, which means that node-schedule will automatically check and update every 0:00 every day.

```
import schedule from 'node-schedule'
import { Activity } from '../models/activity.model.js'

const ON_EVERY_DAY = '0 0 0 * * *'
// const ON_EVERY_DAY = '*/15 * * * *'

export const updateActivityStatus = schedule.scheduleJob(ON_EVERY_DAY, async () => {
    await Activity.updateMany({
        toDate: {
            $lt: new Date()
        },
        {
            status: 'outDate'
        }, { new: true }
    });
});
```

Figure 85 Node-schedule for activity module

```
import schedule from 'node-schedule'
import { Course } from '../models/course.model.js'

const ON_EVERY_DAY = '0 0 0 * * *'
// const ON_EVERY_DAY = '*/15 * * * *'

export const updateCourseStatus = schedule.scheduleJob(ON_EVERY_DAY, async () => {
    await Course.updateMany({
        toDate: {
            $lt: new Date()
        },
        {
            status: 'accomplish'
        }, { new: true }
    });
});
```

Figure 86 Node-schedule for course module

5.10 API document

Education System uses RESTful API, so developers need to write a list of APIs to serve client-side functionality. To let the client-side know what requirements the current API has, what are the parameters to pass, and what are the parameters and responses? In this section, the report provides some APIs for the main modules of the system such as the user module, course module, blog module, activity module. Details of all APIs of the system can be found at the link:

<https://documenter.getpostman.com/view/12892006/TzJu9cyv>

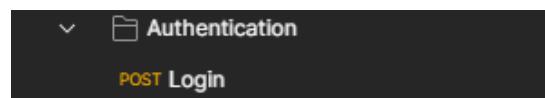


Figure 87 Login API

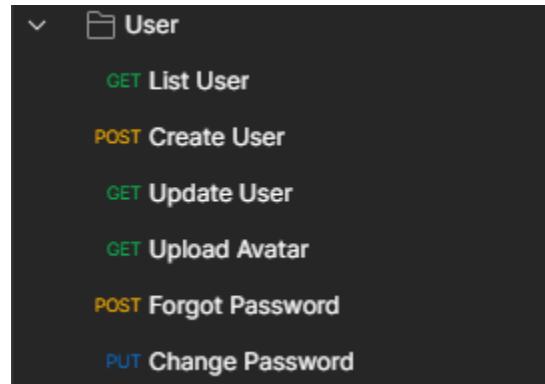


Figure 88 User APIs

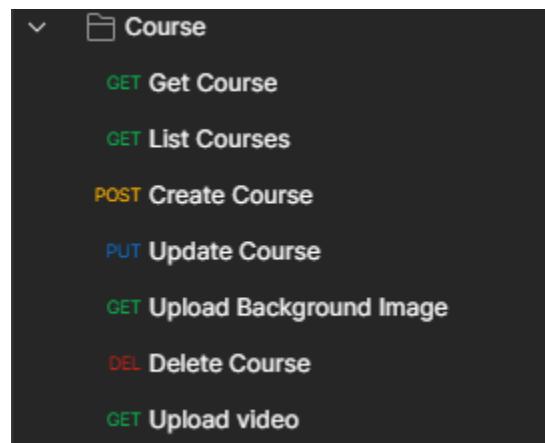


Figure 89 Course APIs

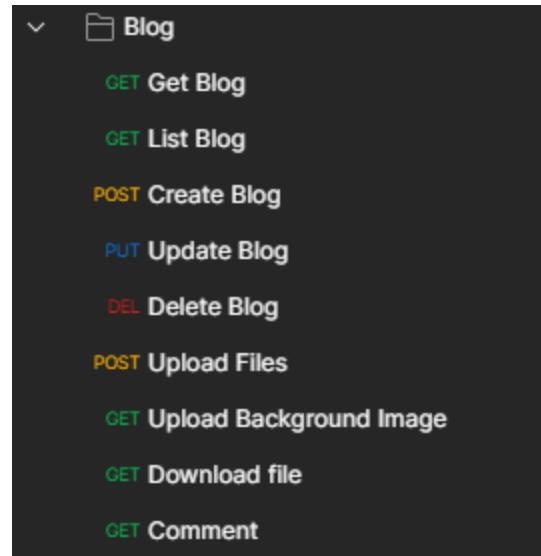


Figure 90 Blog APIs

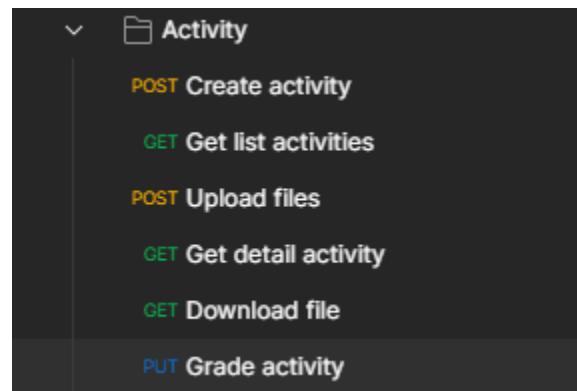
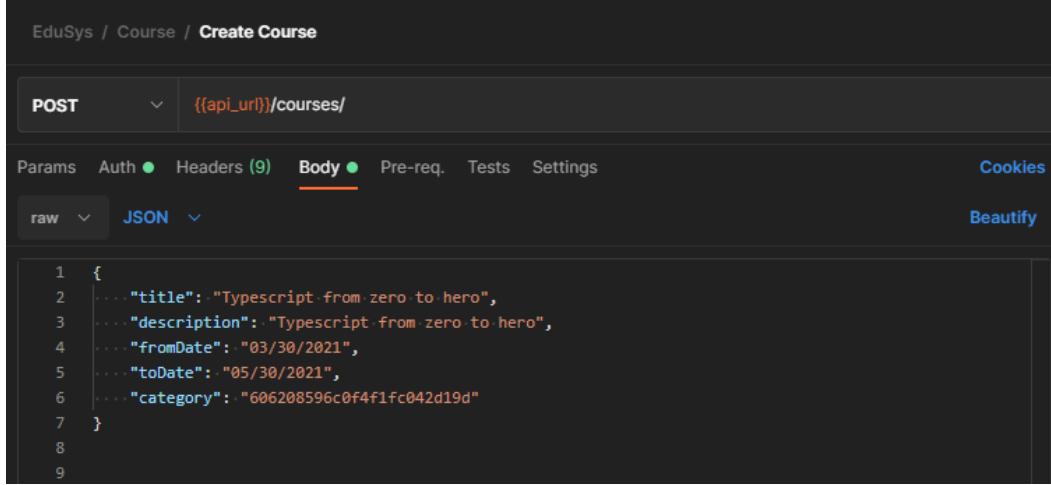


Figure 91 Activity APIs

For example, with an API creating a new course, the API will need course information requests, the creator's information is sent up in the form of a token and sent with the headers.



The screenshot shows the Postman interface with the following details:

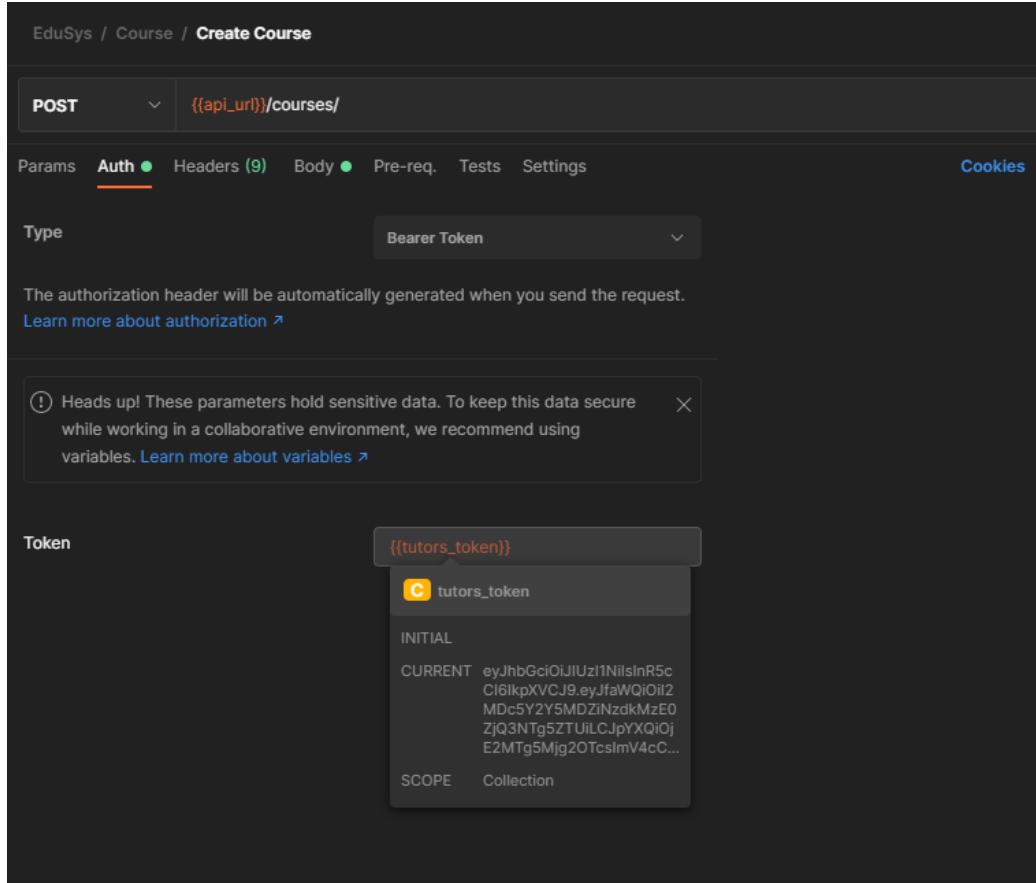
- Method:** POST
- URL:** {{api_url}}/courses/
- Body:** JSON (selected tab)
- Request Body Content:**

```

1  {
2   ... "title": "Typescript from zero to hero",
3   ... "description": "Typescript from zero to hero",
4   ... "fromDate": "03/30/2021",
5   ... "toDate": "05/30/2021",
6   ... "category": "606208596c0f4f1fc042d19d"
7 }
8
9

```

Figure 92 Body of request

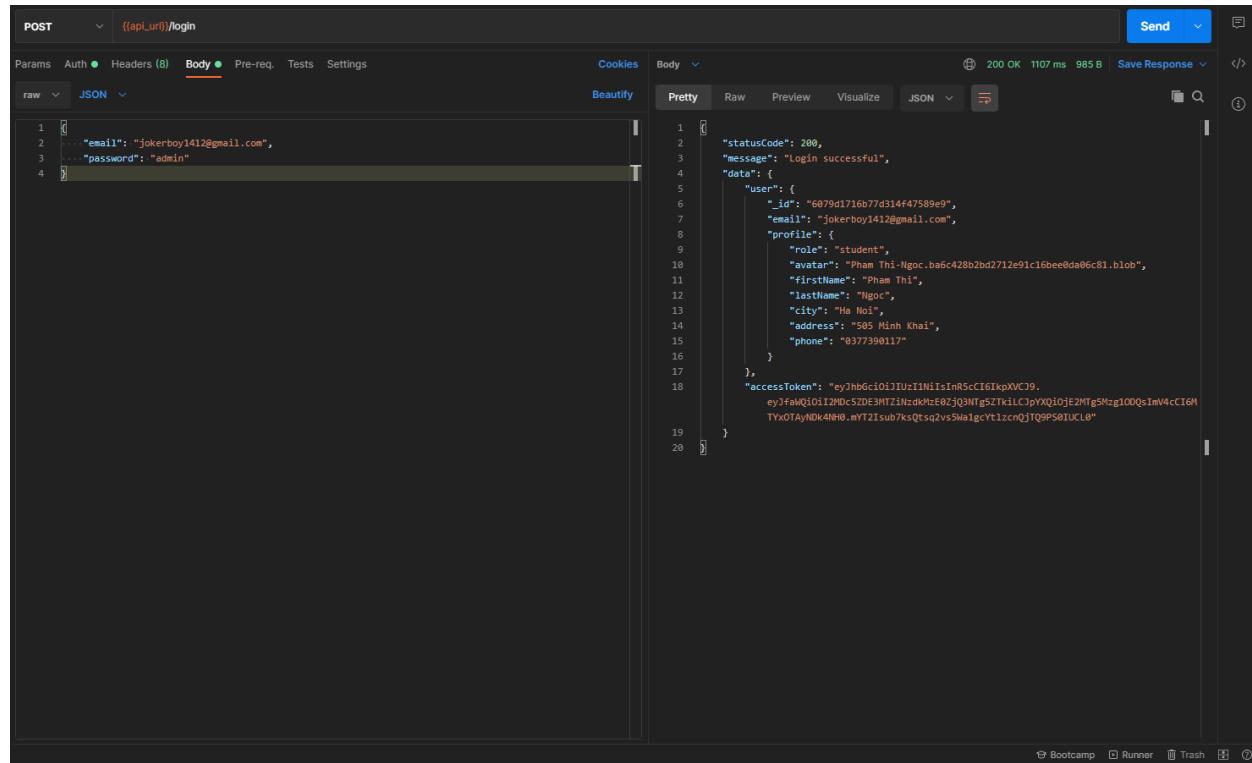


The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** {{api_url}}/courses/
- Auth:** (selected tab)
- Type:** Bearer Token
- Note:** The authorization header will be automatically generated when you send the request.
- Warning:** Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables.
- Token:** {{tutors_token}}
- Token Value:** (A modal window shows the token value: C tutors_token, INITIAL, CURRENT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiJ2MDc5Y2Y5MDZiNzdkMzE0ZjQ3NTg5ZTUlCJpYXQIOjE2MTg5Mjg2OTcsImV4cC...)

Figure 93 Request headers

Each time a user logs in, the system will generate a token based on the encrypted user's information. Each time a user sends a request, depending on each function, token will be sent with headers to be able to authenticate the user's information.



The screenshot shows a POST request to `{{apiUrl}}/login`. The request body contains:

```

1  [
2   ... "email": "jokerboy141@gmail.com",
3   ... "password": "admin"
4 ]

```

The response status is 200 OK, with a response time of 110 ms and a size of 985 B. The response body is:

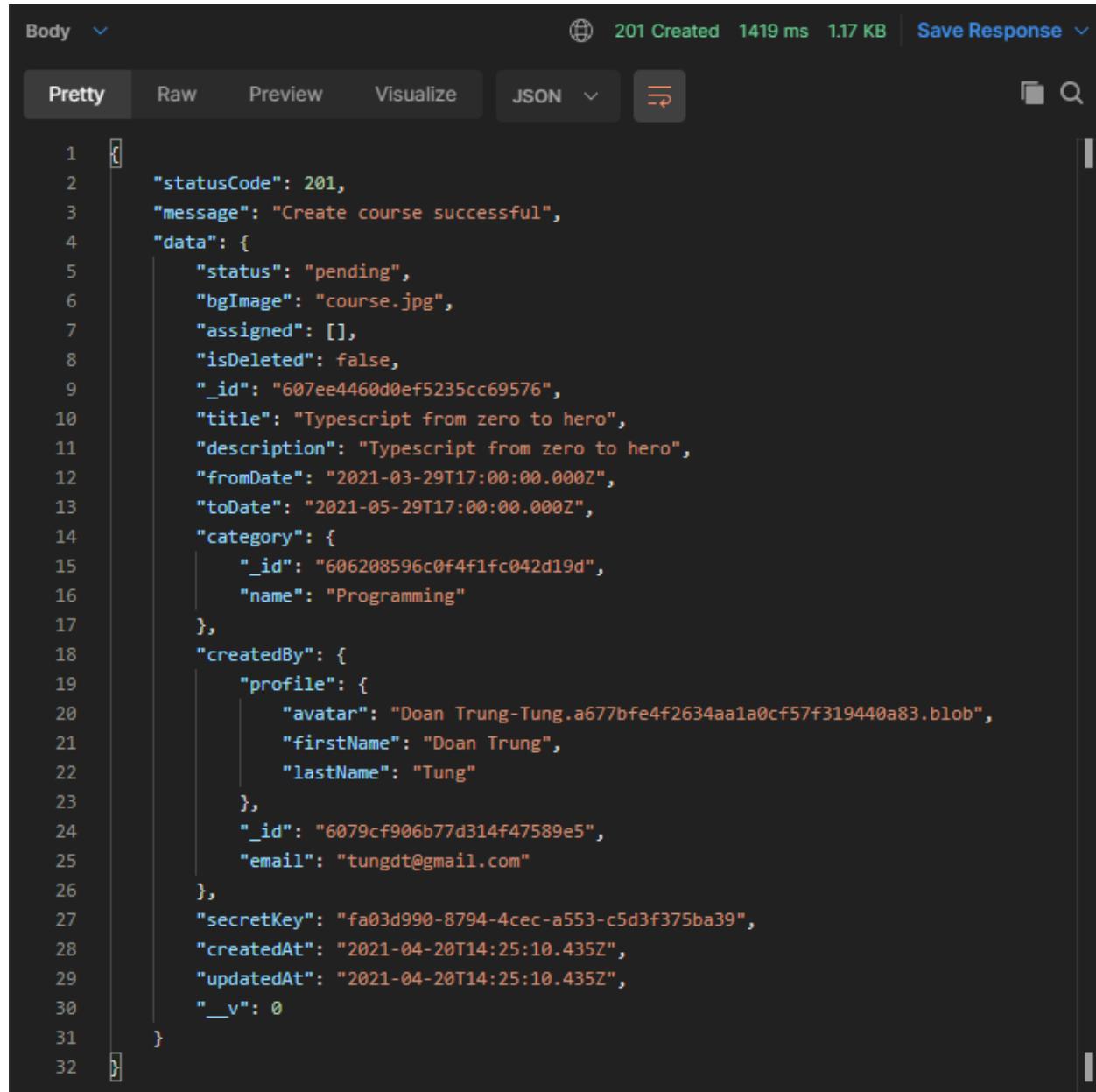
```

1  {
2   "statusCode": 200,
3   "message": "Login successful",
4   "data": {
5     "user": {
6       "_id": "6079d1716b77d314f47589e9",
7       "email": "jokerboy141@gmail.com",
8       "profile": {
9         "role": "student",
10        "avatar": "Pham Thi-Ngoc.b46c428b2bd2712e91c16bee0da06c81.blob",
11        "firstName": "Pham Thi",
12        "lastName": "Ngoc",
13        "city": "Ha Noi",
14        "address": "505 Minh Khai",
15        "phone": "0377390117"
16      }
17    },
18    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXCI9
eyJfaWQiOiI1MDc5ZDE3MTZlNzdkMzE0ZjQ3Nlg5ZTk1LCpYQjOjE2MTg5Mzg1ODQsImV4cCIGh
TYxOTAyNDk4MH0.mYT2Isu07ksQtsg2vs5walgcyt1zcQjTQ9PS0IUCL0"
19  }
20

```

Figure 94 Generate access token

When developers have sufficient information, developers can create a new course on the postman. Results are shown below



The screenshot shows a REST API response in a JSON editor. The status is 201 Created, and the message is "Create course successful". The data returned is a course object with the following details:

```
1  [
2      "statusCode": 201,
3      "message": "Create course successful",
4      "data": {
5          "status": "pending",
6          "bgImage": "course.jpg",
7          "assigned": [],
8          "isDeleted": false,
9          "_id": "607ee4460d0ef5235cc69576",
10         "title": "Typescript from zero to hero",
11         "description": "Typescript from zero to hero",
12         "fromDate": "2021-03-29T17:00:00.000Z",
13         "toDate": "2021-05-29T17:00:00.000Z",
14         "category": {
15             "_id": "606208596c0f4f1fc042d19d",
16             "name": "Programming"
17         },
18         "createdBy": {
19             "profile": {
20                 "avatar": "Doan Trung-Tung.a677bfe4f2634aa1a0cf57f319440a83.blob",
21                 "firstName": "Doan Trung",
22                 "lastName": "Tung"
23             },
24             "_id": "6079cf906b77d314f47589e5",
25             "email": "tungdt@gmail.com"
26         },
27         "secretKey": "fa03d990-8794-4cec-a553-c5d3f375ba39",
28         "createdAt": "2021-04-20T14:25:10.435Z",
29         "updatedAt": "2021-04-20T14:25:10.435Z",
30         "__v": 0
31     }
32 }
```

Figure 95 The response of API creating new course

6 Evaluation and conclusion

6.1 Introduction

In this section, the report will provide an evaluation of the Education System based on a number of criteria such as performance, security, and efficiency after the system has been completed.

6.2 Security

Security is indispensable when developing the website system. To be able to prevent hackers from attacking and stealing user data. Therefore, the Education System project has used a number of encryption features to protect users' information.

The system using JSON Web Token is a method of authenticating users with an encrypted chain based on the user's information so that access to the account can be verified without providing an account and password. This can avoid the user's account information being exposed when they make requests.

```
export const accessToken = (userId) => jwt.sign({ _id: userId }, process.env.JWT_SECRET, { expiresIn: '1d' });
```

Figure 96 Json Web Token based on user information

Each time a user logs in each time, the system will encrypt the user's information and then generate a Json Web Token (JWT) (Figure 97 User login and system will generate the JWT token). Every time a user makes a certain request, the system will check this JWT code to see if the user exists or not (Figure 98 Verify Json Web Token (JWT)).

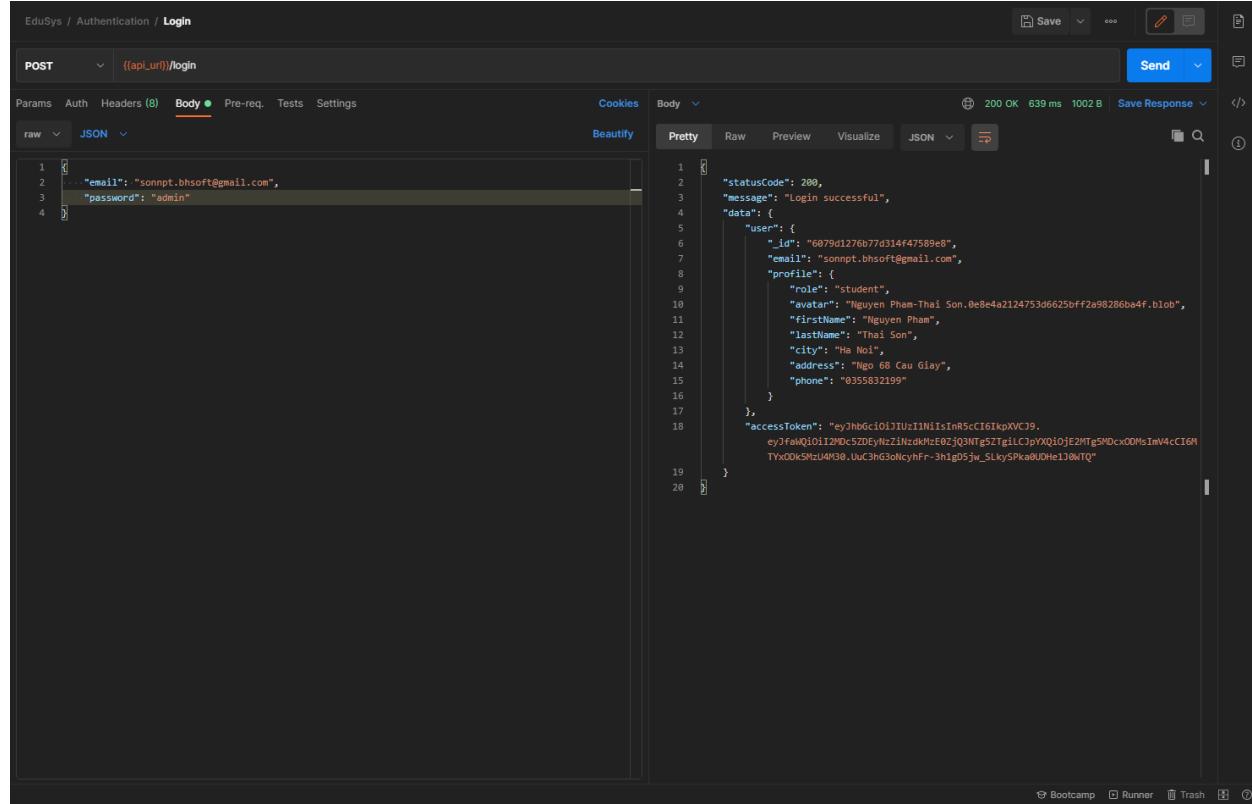


Figure 97 User login and system will generate the JWT token

```

export const verifyToken = async (req, res, next) => {
  try {
    if (!req.headers.authorization && req.headers.authorization.startsWith('Bearer')) {
      return res.status(401).send({ statusCode: 401, message: 'Not authorized', data: {} });
    }

    const token = req.headers.authorization.split(' ')[1];
    const { _id } = jwt.verify(token, process.env.JWT_SECRET);
    const user = await User.findOne({ _id });
    if (!user) {
      return res.status(404).send({ statusCode: 404, message: 'User not found', data: {} });
    }

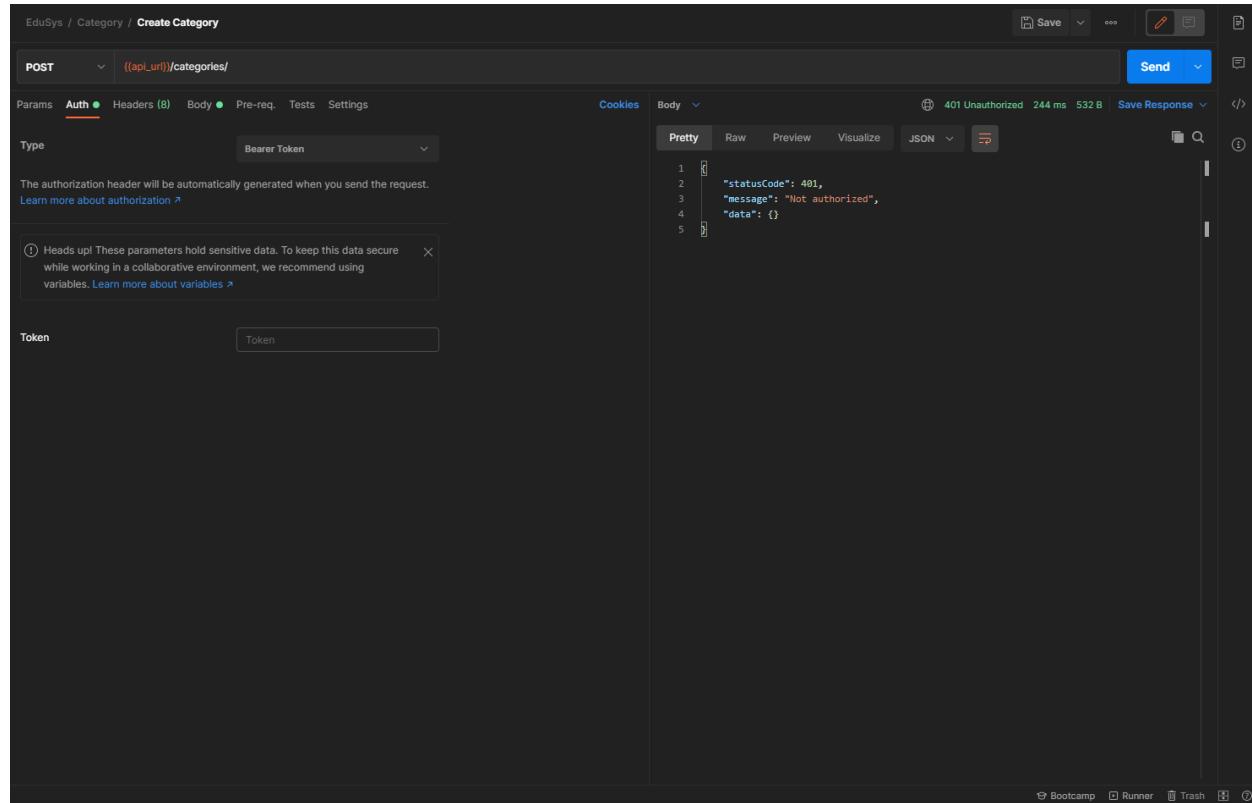
    // assign to req params
    req.user = user;

    return next();
  } catch (error) {
    return res.status(401).send({ statusCode: 401, message: 'Not authorized', data: error });
  }
};

```

Figure 98 Verify Json Web Token (JWT)

In case the user is not logged in, the token expires or the token is not sent when sending the request, the system will report an error with not authorized.



In addition, the user's password is stored in the database is also encrypted, preventing hackers from being able to break into the database and gain user account information. The Education System used bcrypt to encrypt the user's password into a messy string before saving it to the database.

```
UserSchema.pre('save', async function save(next) {
  try {
    if (this.isModified('password')) {
      const salt = await bcrypt.genSalt(10);
      this.password = bcrypt.hashSync(this.password, salt);
    }

    return next();
  } catch (err) {
    return next(err);
  }
});
```

Figure 99 Encrypt the user password

```

_id: ObjectId("6079d26d6b77d314f47589ec")
> profile: object
passwordResetToken: null
isDeleted: false
email: "sonptgch17274@fpt.edu.vn"
password: "$2a$10$70xM3bV5qNLbQh22T8o150W2KrB8a.FsmoiIfw0.pV2H/n5fnBYDiO"
createdAt: 2021-04-16T18:07:43.829+00:00
updatedAt: 2021-04-16T18:07:43.829+00:00
__v: 0

```

Figure 100 Encrypt the user password

Every time a user logs in, the system will compare the password that the user sent with the encrypted password stored in the database.

```

UserSchema.methods.comparePassword = function (password) {
  return bcrypt.compareSync(password, this.password);
};

```

Figure 101 Compare password function

The system also integrates a role-based system, for each function there will be a separate role to perform certain functions. This avoids the user using an account with a student role or tutors to be able to use the admin function.

```

export const validateRole = (role = [] ) => (req, res, next) => !_.includes(role, req.user.profile.role) ?
  res.status(403).send({ statusCode: 403, message: 'Access denied', data: {} }) : next();

```

Figure 102 Validate role

For example, the category-related functions, all of these functions are performed by the administrator (Figure 103 The function of administrator), If a user uses student token or tutors token to use these APIs, the system will issue an error "Access denied" (Figure 104 Access denied)

```

export default (prefix) => {
  prefix.use('/categories', verifyToken, router);

  router.get('/', categoryController.getListCategory);
  router.post('/', validateRole(['admin']), categoryValidator.createCategory, validateRequest, categoryController.createNewCategory);
  router.put('/:id', validateRole(['admin']), categoryController.updateCategory);
  router.delete('/:id', validateRole(['admin']), categoryController.deleteCategory);
};

```

Figure 103 The function of administrator

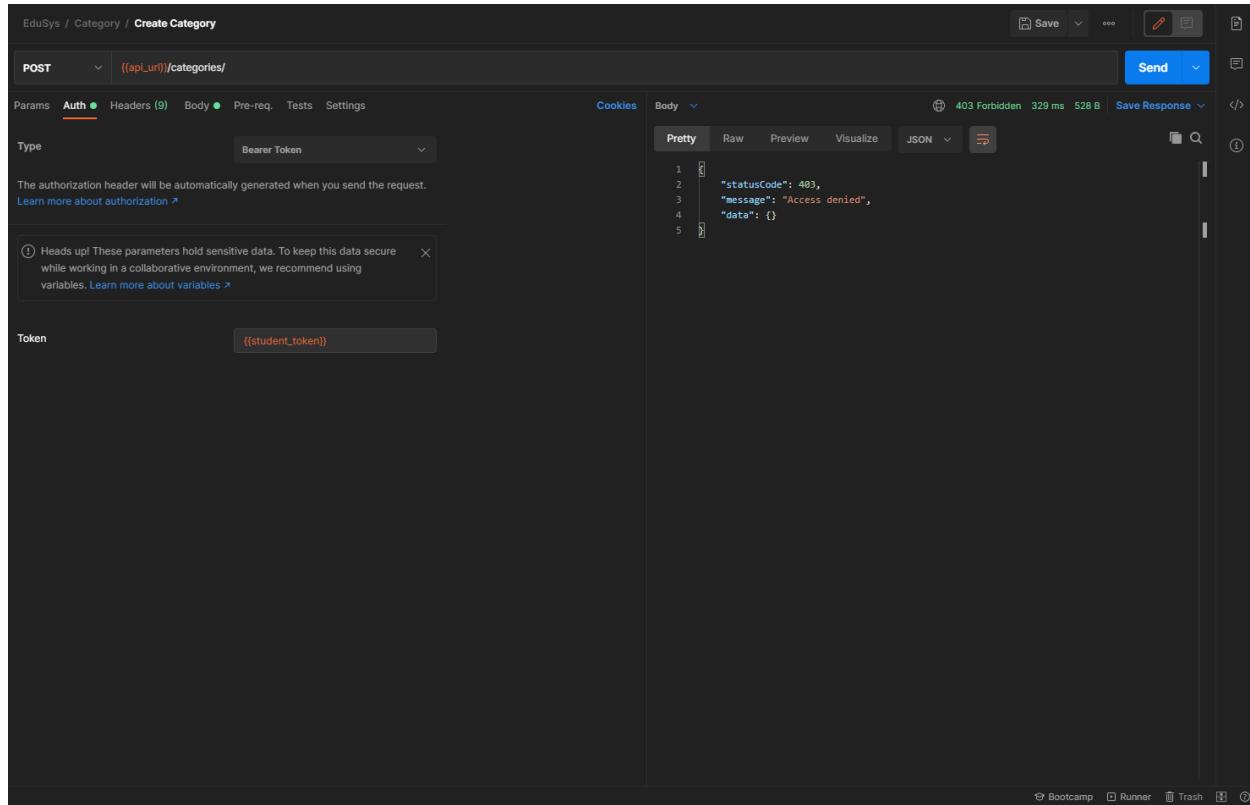


Figure 104 Access denied

6.3 Testing

In this section, the report will provide Education System test cases based on the functions listed in **Appendix C – Education System Functionals**. Details of test cases are listed at **Appendix E – Test Case**

6.4 Evaluation of Education System

Based on the analysis of requirements and blueprints presented in the section **Requirement Analysis** and **Software Design**. The Education System has fully completed the functions that have been analyzed above. This section reports on the evaluation of functions for roles.

6.4.1 Administrator

The functions of the administrator have been fully implemented, including viewing statistical information on the number of total users of the system, the number of categories, the number of courses. The administrator can keep track of the number of courses that have been approved, the number of courses rejected, and the number of completed courses in a visual chart. In

addition, the administrator can manage information such as adding, modifying and deleting all users in the system, managing categories, and managing courses created by tutors.

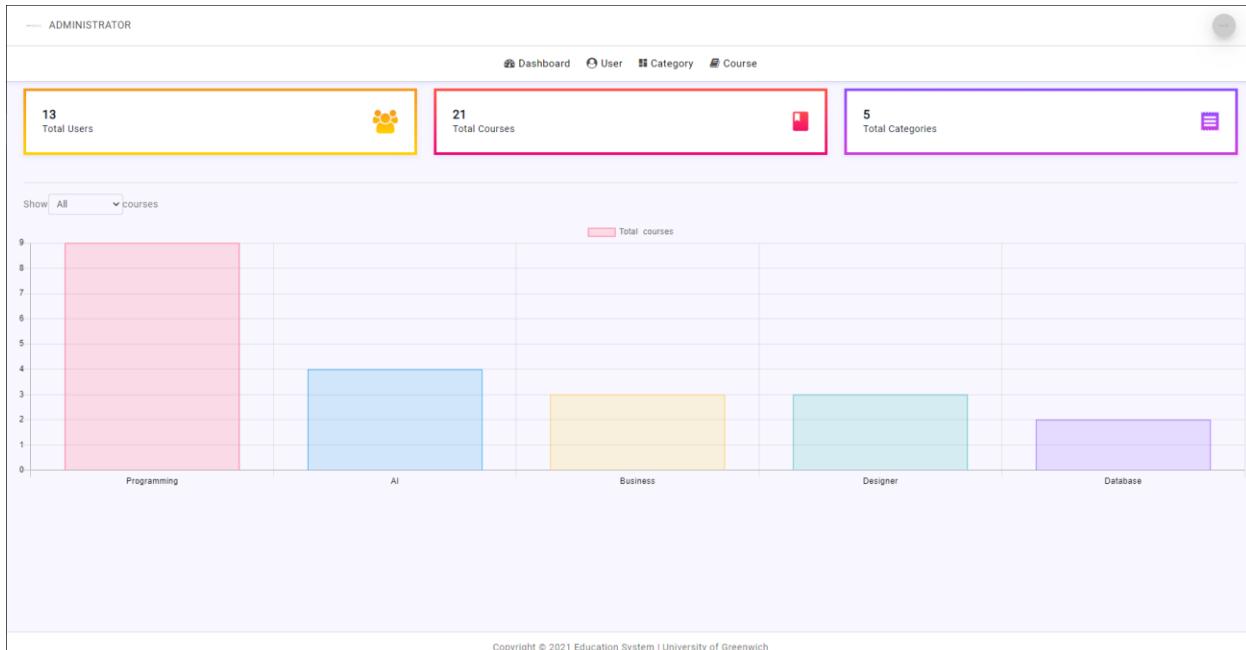
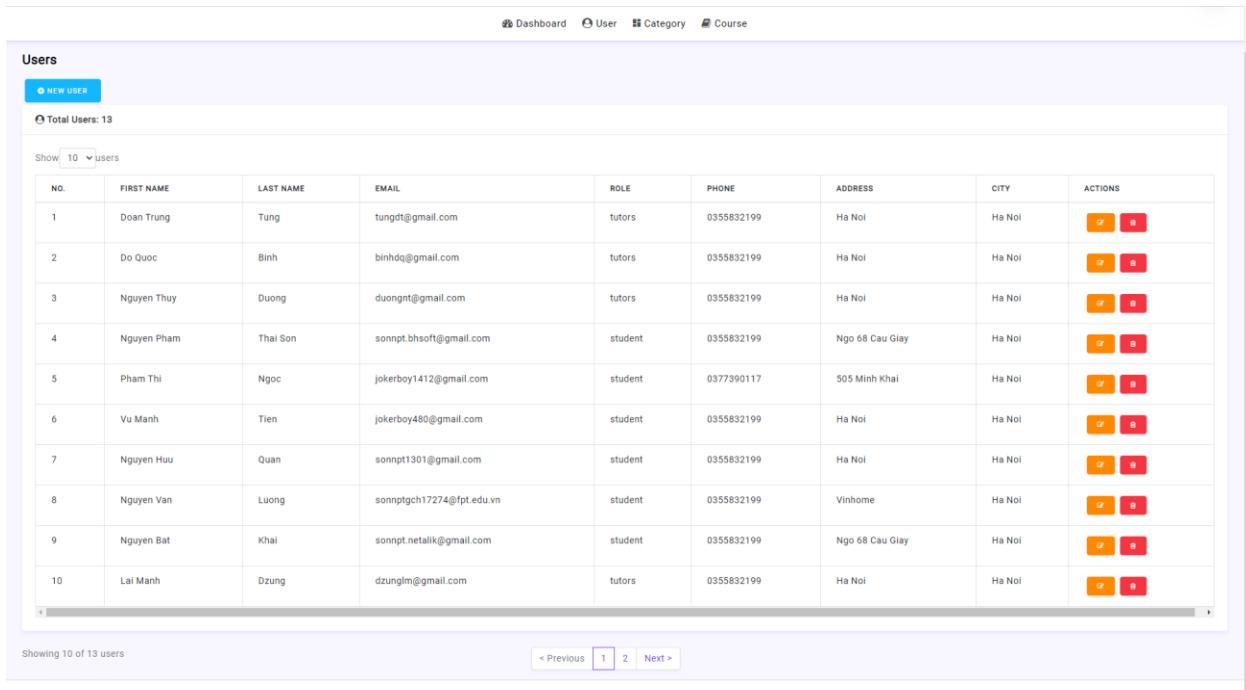


Figure 105 Dashboard page



The Manage users page lists the following users:

NO.	FIRST NAME	LAST NAME	EMAIL	ROLE	PHONE	ADDRESS	CITY	ACTIONS
1	Doan Trung	Tung	tungdt@gmail.com	tutors	0355832199	Ha Noi	Ha Noi	
2	Do Quoc	Binh	binhdq@gmail.com	tutors	0355832199	Ha Noi	Ha Noi	
3	Nguyen Thuy	Duong	duongnt@gmail.com	tutors	0355832199	Ha Noi	Ha Noi	
4	Nguyen Pham	Thai Son	sonnpt.bhsoft@gmail.com	student	0355832199	Ngo 68 Cau Giay	Ha Noi	
5	Pham Thi	Ngoc	jokerboy1412@gmail.com	student	0377390117	505 Minh Khai	Ha Noi	
6	Vu Manh	Tien	jokerboy480@gmail.com	student	0355832199	Ha Noi	Ha Noi	
7	Nguyen Huu	Quan	sonnpt1301@gmail.com	student	0355832199	Ha Noi	Ha Noi	
8	Nguyen Van	Luong	sonnptgch17274@fpt.edu.vn	student	0355832199	Vinhome	Ha Noi	
9	Nguyen Bat	Khai	somppt.netalk@gmail.com	student	0355832199	Ngo 68 Cau Giay	Ha Noi	
10	Lai Manh	Dzung	dzunglm@gmail.com	tutors	0355832199	Ha Noi	Ha Noi	

Showing 10 of 13 users

< Previous 2 Next >

Copyright © 2021 Education System | University of Greenwich

Figure 106 Manage users

— ADMINISTRATOR

Dashboard User Category Course

Category

[NEW CATEGORY](#)

Total Categories: 5

NO.	NAME	ACTIONS
1	Programming	 
2	Designer	 
3	Database	 
4	AI	 
5	Business	 

Copyright © 2021 Education System | University of Greenwich

Figure 107 Manage categories

List courses

Total Courses:

Show	All	courses	Show	10	courses	
NO.	TITLE	CATEGORY	TUTORS	DURATION	STATUS	ACTIONS
1	Introductory C Programming	Programming	Doan Trung Tung	17-Apr-2021 - 21-Apr-2021	Accomplish	edit
2	Beginning C++ Programming - From Beginner to Beyond	Programming	Doan Trung Tung	17-Apr-2021 - 08-May-2021	On process	edit
3	Learn and Understand NodeJS	Programming	Doan Trung Tung	17-Apr-2021 - 15-Jul-2021	On process	edit
4	Web Design for Beginners: Real World Coding in HTML & CSS	Programming	Lai Manh Dzung	17-Apr-2021 - 29-May-2021	On process	edit
5	React - The Complete Guide (incl Hooks, React Router, Redux)	Programming	Lai Manh Dzung	17-Apr-2021 - 05-Jun-2021	On process	edit
6	For Free - Deploy Quickly Spring Boot on Heroku With MySQL	Programming	Do Quoc Binh	17-Apr-2021 - 27-May-2021	On process	edit
7	Distributed Systems & Cloud Computing with Java	Programming	Do Quoc Binh	14-Apr-2021 - 01-Jul-2021	On process	edit
8	Artificial Intelligence A-Z™: Learn How To Build An AI	AI	Nguyen Thuy Duong	17-Apr-2021 - 29-May-2021	On process	edit
9	Modern Artificial Intelligence with Zero Coding	AI	Nguyen Thuy Duong	17-Apr-2021 - 21-Jul-2021	On process	edit
10	Artificial Intelligence 2018: Build the Most Powerful AI	AI	Nguyen Thuy Duong	17-Apr-2021 - 15-Jun-2021	On process	edit

Showing 10 of 21 courses

< Previous 1 2 3 Next >

Copyright © 2021 Education System | University of Greenwich

Figure 108 Manage courses

On the admin management pages, the system provides a number of filters to help the admin filter data and find information quickly. In addition, the system provides a pagination feature that makes it easier for the admin to manage information.

6.4.2 Tutors

The tutors' functionality has been fully implemented. The functions of tutors include adding and editing course information. In each course, tutors can manage all blogs posted by students, tutors can also post blogs. In addition, the tutors can manage all activities in each course, view all student submissions and grade student's submissions.

EduSys 

[Home](#) [Categories](#) [About](#)

 Course Detail

Categories / Programming / Introductory C Programming

Introductory C Programming

 Doan Trung Tung

Duration: 17-Apr-2021 - 21-Apr-2021

Category: Programming

Total students: 10

 COURSE  BLOG  ACTIVITY

Description

Learn Essential Programming Fundamentals. Master programming skills to solve complex problems.



Copyright © 2021 Education System | University of Greenwich

Figure 109 Manage Course

List students 

1. Pham Thi Ngoc

Figure 110 View list students in the course

Waiting for approving (0) 

No blog

Figure 111 Manage student's blog

New blog X

TITLE *

CONTENT *

BACKGROUND IMAGE

No file chosen

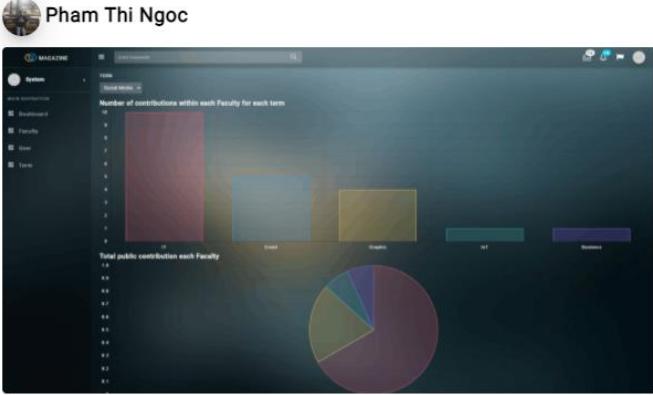


Default background image

Figure 112 Create new blog

Blog detail

X



Pham Thi Ngoc

Introduce the chart package

5 days ago

In order to build a chart, you can go to react-chartjs-2 at the below link
<https://www.npmjs.com/package/react-chartjs-2>

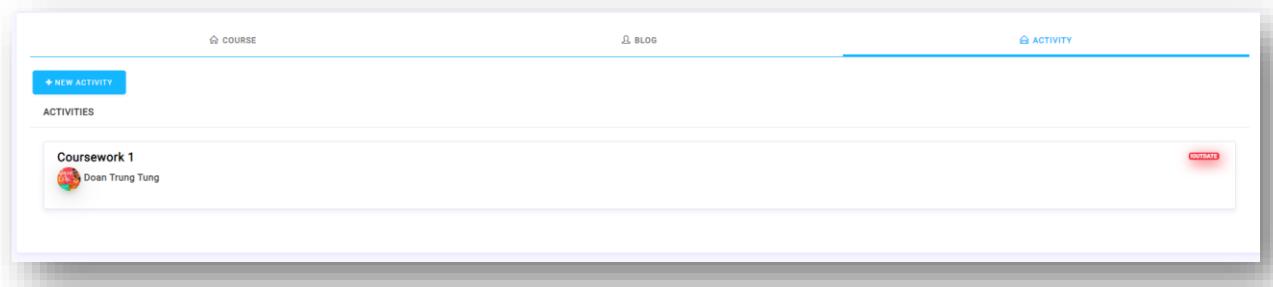


Doan Trung Tung
 Thanks for sharing!!!

5 days ago

Write comment here...

Figure 113 Comment on student's blog



The screenshot shows the Moodle course management interface. At the top, there are three tabs: COURSE, BLOG, and ACTIVITY. The ACTIVITY tab is active. Below the tabs, there is a button labeled '+ NEW ACTIVITY'. Under the ACTIVITIES section, there is a card for 'Coursework 1' submitted by 'Doan Trung Tung'. There is also a small red button labeled 'GRADE'.

Figure 114 Manage activities in course

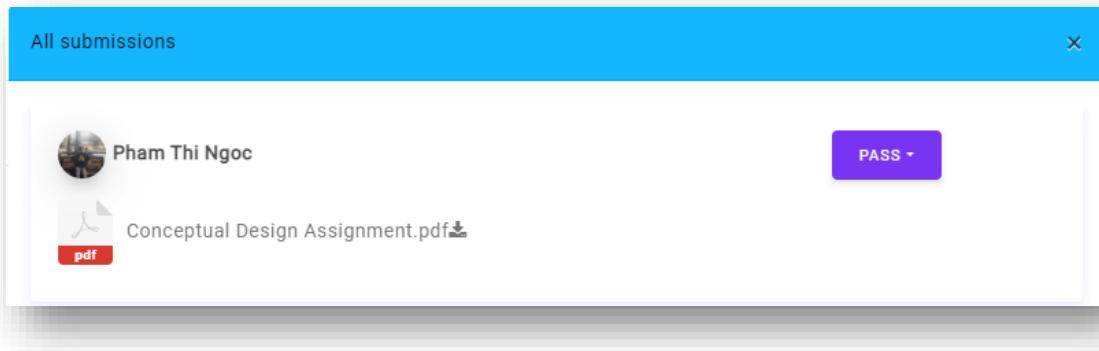


Figure 115 View all student's submission and grade

6.4.3 Student

Students can join courses by submitting a request to join the courses, after which students can receive a key to access the class via email. In each course, students can post blogs and be approved by tutors in each course. In addition, students can join activities and submit individual assignments/courseworks for each course. Students can download the uploaded tutor's requirement files and view their assignments/courseworks grades.

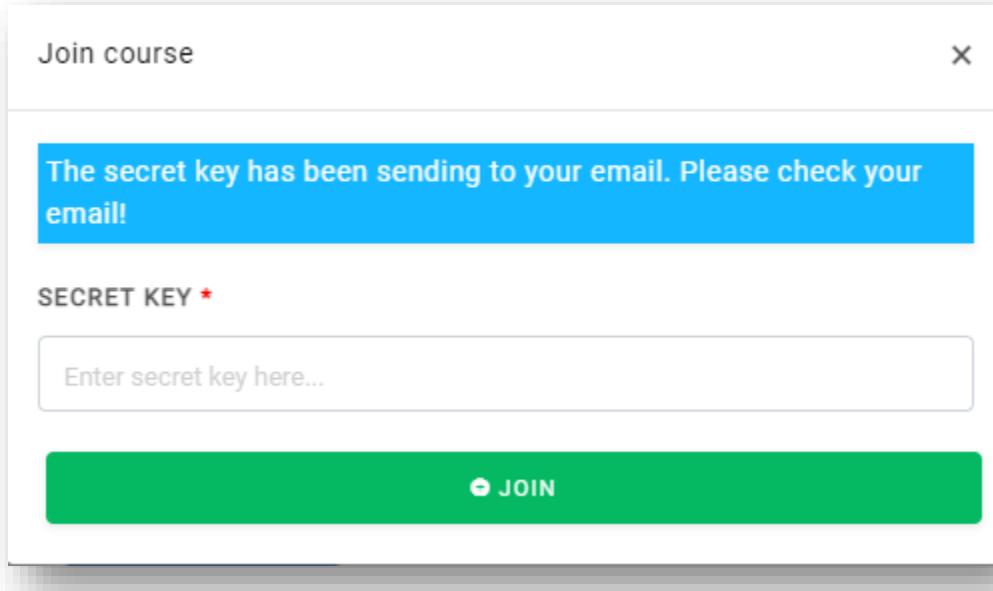


Figure 116 Join course

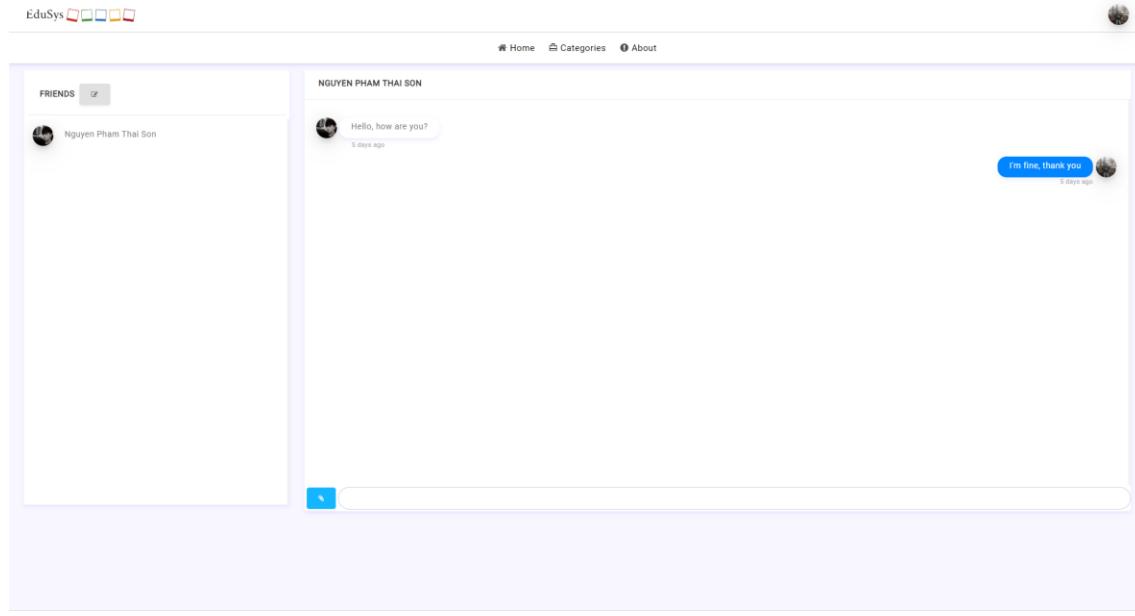
Activity Detail X

COURSEWORK 1 OUT DATE

CREATED BY	 Doan Trung Tung
DESCRIPTION	Limit 5000 words
DEADLINE	17-Apr-2021 - 20-Apr-2021
GRADE	PASS
REQUIREMENTS	 coursework.docx DOCX
FILE SUBMISSIONS	Can't submit

Figure 117 Submit file and view grade

In addition, the system integrates chat feature, allowing students with tutors to chat with each other with the realtime chat system.



The screenshot shows a social networking interface for 'EduSys'. On the left, there's a 'FRIENDS' section with a 'Nguyen Pham Thai Son' profile. The main area is a chat window titled 'NGUYEN PHAM THAI SON' where two users are messaging. The first message is from 'Nguyen Pham Thai Son' at '5 days ago': 'Hello, how are you?'. The second message is from an unnamed user at '5 days ago': 'I'm fine, thank you'. The footer of the page includes a copyright notice: 'Copyright © 2021 Education System | University of Greenwich'.

Figure 118 Chat system

6.5 Conclusion

6.5.1 Lessons learnt

Through this project, a number of technologies have been learned to apply to the project, helping the system to have more practical features. Firstly, learn how to use third-party services such as sending mail through Google service, s3 backup service provided by Amazon Web Services. Second, learn the basics of sockets to build a simple real-time chat system. Thirdly, learn how to use nodes-schedule package to be able to schedule some time functions automatically in the system.

In addition to the technologies learned and applied to the project, the reporting style has also been greatly improved compared to other reports written before. The current paper has been written more professionally, including references to Havard references to increase transparency and convincing of each thesis.

6.5.2 Problems / difficulties

After completing the project, the system still has some problems that have not been really resolved due to a lack of ideas and lack of experience in the process. Below are some of the system problems currently encountered.

- Components used in the system are not really diverse, using too many modals instead of individual pages.
- Sockets using systems are not really optimal when they are not properly divided into separate controllers.
- The interface is still simple but not really eye-catching, the colors used in the system are not too much, including only two basic colors: white for the interface and blue for the buttons.
- The interface also has some small responsive bugs
- The system status messages are still not displayed correctly or the function is not being performed.

- In addition, some functions designed and planned from the beginning have not been implemented, such as assign other tutors in the course, leave the course function for the student, manage the student leaving the course.
- Chat system does not have a pagination feature. In the case of the site had a large number of users with each chat up to several million messages per user. At a certain time, if a series of users send requests to get all of the message information, this could lead to server processing slow down or even cause server death due to too many requests simultaneous

In the process of building the system, developers cannot avoid encountering errors during the execution of any project. Most of the problems that happened during the project implementation, the majority of them can be overcome, but they are still not the best solution and there are some problems that have not really been resolved in this project. Problems and solutions during project implementation are listed below.

Table 3 List bugs

No	Bugs	Solution
1	Can't filter blogs by user cause by missing filter by createdBy in queryBuilder function	Additional filter by createdBy in queryBuilder
2	Can't filter blogs by status cause by missing filter by status in queryBuilder function	Additional filter by status in queryBuilder
3	Can't upload images/files to s3 cause by provide wrong access key and secret key	Modify the access key and secret key
4	Client: The input file does not trigger / does not appear immdiatly after choosing the file	Users have to re-open the modal, the file could be appeared
5	Client: The input file is not clear after uploading the file success	This bug cannot be resolved temporarily. The temporarily solution is user have to refresh the browser
6	Client: When join a course failed, if join another course, reducer still save error. Therefore, if access another	Users have to refresh the browser to remove error

	course already join -> page still send join request because error still save in reducer	
7	Client: When the data is loading, the data is displayed as undefined	Add more condition if data is not existed with blank template string

6.5.3 Future improvements

In the future, if given the opportunity, issues or functions that have not been implemented in the project will be improved as listed below.

- Re-structuring directory using sockets, dividing the socket using the correct individual controller, instead of writing the entire socket file to a common.
- Improved user interface.
- Reduce the use of too many modals.
- Implemented additional features that can be assigned to other tutors posting and can modify course information.
- Implement the function of leaving the course for students, managing the students leaving the course for tutors.
- Add features related to admin, for example, can edit the logo, banner, main color of the system.
- Add the ability to send statistics in PDF format via Gmail to admin on a monthly or yearly basis.
- Added online payment feature for paid courses
- Add pagination feature for messaging features like Facebook, each time the user scrolls up will make a request to be able to load the previous messages instead of loading them all.
- Added the feature of displaying push notifications so that users can receive notifications immediately instead of receiving email notifications.

7 Conclusion

After completing my year-end graduation project, during my studies at school, I learned a lot. Initially, I learned how to search scholarly literature, learn content from academic sources, and how to write a report professionally, how to use the resources citations from scholarly sources in the form of Harvard references added to the article increased transparency and support for each answer. In addition, I also learned how to do a project to develop a website system. In which, how to apply software development lifecycle models is reasonable for each project. Next, I also learned how to design diagrams including models such as Usecase diagram, flow chart, erd diagram... to help build the system from the ground up to help me develop features easily important and necessary. After learning to program, I was also able to build a relatively complete system compared to what was planned from the start. However, the system is still not perfect as I expected. In the future, I will try to develop my skills more and make better quality systems.

References

- Alan, D., Barbara, W. H. & Roberta, R. M., 2012. *SYSTEM ANALYSIS AND DESIGN*. 5th ed. United States of America: John Wiley & Sons, Inc.
- Alex, B. & Eve, P., 2017. *Learning React*. 1st ed. United States of America: O'Reilly Media.
- Blaschke, L. M., Hase, S. & Kenyon, C., 2014. *Experiences in self-determined learning*. Africa: s.n.
- Carlos, C., Steven, M. & Peter, R., 2011. *Database Systems: Design, Implementation, and Management, Ninth Edition*. 9th ed. United States of America : Joe Sabatino.
- Ethan, B., 2016. *Learning JavaScript*. 3rd ed. United States of America: O'Reilly Media.
- Hector, M. G., Jeffrey, U. D. & Jennifer , W., 2002. *DATABASE SYSTEMS The Complete Book*. 2nd ed. U.S: Pearson Education, Inc.
- Islam, M., 2019. *How I learn any front-end framework*. [Online]
Available at: <https://dev.to/imm9o/how-i-learn-any-front-end-framework-29a2>
[Accessed 20 December 2020].
- Jon, P. A. & Michele, D. E., 2007. *Learning PHP and MySQL*. 2nd ed. United States of America.: O'Reilly Media.
- Kristina, C., 2013. *MongoDB: The Definitive Guide*. 2nd ed. United States of America: O'Reilly Media.
- Leon, S. & Richard, R., 2003. *Web Application Architecture Principles, Protocols and Practices*. USA: John Wiley & Sons Ltd.
- Robin, N., 2014. *Learning PHP, MySQL, JavaScript, CSS & HTML5*. 3rd ed. United States of America: O'Reilly Media.
- Seyed , T. M. ". & Hugh , W. E., 2006. *Learning MySQL*. 1st ed. United States of America: s.n.
- Shikha , M. & Dinesh, J. C., 2012. International Journal of Advanced Research in Computer Science and Software Engineering. In: *A Comparative Analysis of Different types of Models in Software Development Life Cycle*. s.l.:Jaunpur: Divyendu Kumar Mishra, pp. 285-290.

Shyam, S., 2018. *Angular: Up and Running*. 1st ed. United States of America: O'Reilly Media.

Appendix A – Project Proposal

1. Introduction

Currently, there are many schools or any other educational institution still manage their schools in a common way on paper such as manage paper-based tests, taking attendance through attendance book, schedule and instructional schedule are still communicated verbally. Significant announcements are also only communicated through the monitor's class.

Therefore, I want to develop a website education system, which makes it easier and more convenient for schools to manage specific tasks for each department than the common way. Teachers, as well as students, can easily keep track of their teaching and class schedules at any time through their phones or computers with an internet connection. The training department is also easier to manage subjects, schedule teachers in the school. For tests or student reports, when submitted, they will be stored on the school database to avoid the usual loss of paper-based tests. Students as well as parents can easily access the school website to keep track of their grades.

Here are some technologies that I can use in developing my website system.

I'll use React to develop front-end. "ReactJS – An open-source JavaScript Library for Front-end Development" (Naimul, 2017). React is currently very popular to build on the UI side that delivers good response speed when user inputting. The components were originally developed for internal use by Facebook and later became open source and added functionality by many contributors. In addition to Facebook, a number of other organizations are also using ReactJS and React Native like: Instagram, Netflix, WhatsApp...

Some of the benefits of ReactJS: easy to use, reusable components, quick response when displaying the results of large amounts of data in components, easier to debug with Chrome extension, using virtual DOM to help optimize the page better ...

About back-end, Express JS is one of the most popular frameworks in NodeJS today, providing powerful features for website or mobile development. Express JS supports HTTP methods and middleware makes APIs easy to use, helps to shorten the application development time greatly. Some of the main functions of Express JS include:

- Define middleware to support for organizing and reusing code
- Define routes and requests method to server easily
- Allows to return HTML based on parameters
- Support to create REST API server
- Easy to connect with database such as MongoDB, MySQL...

To be able to manage website data, I choose MongoDB to manage my website data.

“MongoDB is an open source database management system (DBMS) that uses a document-oriented database model that supports numerous kinds of data. Rather than using tables and rows as in relational databases, the MongoDB architecture is formed of collections and documents. MongoDB stores data in JSON-like documents that can vary in structure.” (Decode , 2019)

2. Aim

The aim of this project is to develop a website system for schools. This system will make it easier for all school departments to manage tasks. In addition, students are also subject of this system, they can log into this system to track their class schedule as well as track their academic results.

3. Key phrases

Education website, educational, manage school system, website educational system, manage school system

4. Objective(s)

A. *To gather the requirements for the website educational system*

Activities:

- Analyze the requirements
- Write the report describe the specific functions for administrator, training department, teacher, student

Deliverables:

- ✓ The report describes detail the functions for administrator, training department, teacher, student

B. To analyze requirements and draw diagrams that represent the functions, operations, and relationships of requirements

Activities:

- Design the relationship between the necessary objects: Administrator, Training Department, Teacher, Category, Account, Course, Class ...
- Design flow chart demonstrates the functions for each object: Administrator, Training Department, Teacher, Student.
- Design use case diagram demonstrates the activities for each object: Administrator, Training Department, Teacher, Student.

Deliverables:

- ✓ ERD diagram
- ✓ Flow chart
- ✓ Use case diagram

C. To design the website educational system

Activities:

- Plan to design user interface for Administrator, Student, Training Department, Teacher
- Plan to implement the functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for students...
- Choose methods, technologies using build the system

Deliverables:

- ✓ Wireframe
- ✓ The report on the reasons for choosing methods, technologies.

D. To implement the website educational system

Activities:

- Implement user interface for Administrator, Student, Training Department, Teacher pages
- Design database for Administrator, Student, Teacher, Training Department, Course, Class, Category...
- Implementation of functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for students...

Deliverables:

- ✓ A completed the website educational system.

E. To test the entire website educational system

Activities:

- Test the functions of website: add, update, delete account of administrator, add, update, delete category, course and arrange class schedules of training department...

- Evaluate the system through the result of test cases
- Write the report about the problems need to be improved or solved

Deliverables:

- ✓ The report test cases
- ✓ The report evaluates the system through the result of test case
- ✓ The report about the problems need to be improved or solved

F. To evaluation the entire website educational system

Activities:

- Create a survey to collect evaluation of user about the system
- Evaluate the result of survey

Deliverables:

- ✓ Screenshot the result of survey
- ✓ The report evaluates the result of survey

G. Conclusion

Activities:

- Evaluation the strength and weakness the entire system
- Write the report about the plan for what need to improve in the future

Deliverables:

- ✓ The report about the strength and weakness of the system
- ✓ The report about the plan for what need to improve in the future

H. Write the final report

Activities:

- Write the final report

Deliverables:

- ✓ The final report

5. Legal, Social, Ethical and Professional

A. Legal

For any system, there will be large amounts of critical user data that need to be secured. With the rapid development of the internet, this important information is easy for hackers to steal information, hack the system, breach information, internal sell user information, which will affect the reputation the credit of the business. These issues belong to legal procedures. Therefore, developers need to ensure that the protection of user information, as well as system security, comply with the ethical rules, the information technology law. Currently, there are a lot of pertinent laws that are applied by different governments of several countries such as: Computer Fraud and Abuse Act of 1986 (CFA Act), National Information Infrastructure Protection Act of 1996, USA Patriot Act of 2001, Communications Decency Act of 1996 (CDA). (Michael R. Bartolacci et al, 2014). Compliance with ethical codes of conduct will help protect IT and ICT elements from any form of abuse.

Education website will ensure the legal as follow:

- User information will be kept confidential and protected
- When it is necessary to collect user information, including each department such as teachers, students... will have a specific reason.
- Don't use user information for personal purposes
- Do not interfere with the user's computer

B. Social

The social issues addressed for any system can also directly affect many specific areas outside of real society, can also affect countries, cultures, people, and human trends in society. Specifically, certain businesses use wrong information and spread false information about other countries politically or criticize the culture, lifestyle ... and are widely spread on social networks this could be the reason for conflict between nations. Poor information security issues lead to hackers to steal users' information, the information sold by the hacker will cause trouble to those users outside of society as being harassed even financial fraud leads to an impact on their lives outside of society. In addition, the system uploads inappropriate content such as images, violent videos, lashes, racism, inappropriate images, or posts prohibited products under the law. All of the above not only directly affect the business but also indirectly affect many issues inside and outside the society.

Therefore, we ensure that the content posted on the school website is consistent with school criteria, excluding images, negative content, violence, racism... to avoid influencing to the social life of students at school and in society.

C. Ethical

Currently many information technology experts have created a term called Ethical Hacking (Bernd Carsten Stahl et al, 2014), which means that hackers will break into a certain system and try to find security holes, data systems. information and information to notify such businesses so that they can promptly remedy such gaps. Nowadays, many businesses also hold competitions for hackers to find them vulnerable to their security and pay people to find them. However, if certain businesses intentionally break into the systems of other businesses, they will violate the internal ethics of the business, if they intentionally use the user's information for the wrong purpose, sell. They are out, that business has violated the morality.

To ensure ethics:

- Provide necessary warnings and notices when users perform any payment-related functions or may pose a danger to users.
- Do not post inappropriate content to the learning environment

- With mainstream lectures provided by the school, content is guaranteed not to be copied or taken from outside the school

D. Professional

(R.Mullen et al, 2015) Information confidentiality should be disseminated to businesses, and ethical and confidentiality issues need to be thoroughly educated and trained. Businesses need the skills to be able to deal with data and computer security issues, which can help businesses save resources and risk from problems hacker attacks system.

To have a professional website we will need to keep in mind the following:

- Ensuring high privacy in user data / information
- Exercise caution in modifying systems or data
- Regularly collecting user feedback improves the system when needed

6. Plan

	Task Name	Duration	Start	Finish
1	▪ Website Educational System Project	194 days?	Sat 8/8/20	Fri 4/30/21
2	▪ To gather the requirements for the website educational system	7 days	Sat 8/8/20	Mon 8/17/20
3	Analysis the requirements	5 days	Sat 8/8/20	Thu 8/13/20
4	Write the report describe the specific functions for administrator, training department, teacher, student	2 days	Fri 8/14/20	Mon 8/17/20
5	▪ To analyze requirements and draw diagrams that represent the functions, operations, and relationships of requirements	16 days	Tue 8/18/20	Tue 9/8/20
6	Design the relationship between the necessary objects: Administrator, Training Department, Teacher, Category, Account, Course, Class ...	6 days	Tue 8/18/20	Tue 8/25/20
7	Design flow chart demonstrates the functions for each object: Administrator, Training Department, Teacher, Student	5 days	Wed 8/26/20	Tue 9/1/20
8	Design use case diagram demonstrates the activities for each object: Administrator, Training Department, Teacher, Student	5 days	Wed 9/2/20	Tue 9/8/20
9	▪ To design the website educational system	32 days	Thu 9/10/20	Fri 10/23/20
10	Plan to design user interface for Administrator, Student, Training Department, Teacher	10 days	Thu 9/10/20	Wed 9/23/20
11	Plan to implement the functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for student	20 days	Thu 9/24/20	Wed 10/21/20
12	Choose methods, technologies using build the system	2 days	Thu 10/22/20	Fri 10/23/20
13	▪ To implement the website educational system	90 days	Sat 10/24/20	Tue 2/23/21
14	Implement user interface for Administrator, Student, Training Department, Teacher pages	30 days	Sat 10/24/20	Tue 12/1/20
15	Design database for Administrator, Student, Teacher, Training Department, Course, Class, Category...	10 days	Wed 12/2/20	Tue 12/15/20
16	Implementation of functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for students...	50 days	Wed 12/16/20	Tue 2/23/21
17	▪ To test the entire website educational system	14 days	Wed 2/24/21	Mon 3/15/21
18	Test the functions of website: add, update, delete account of administrator, add, update, delete category, course and arrange class schedules of training department...	7 days	Wed 2/24/21	Thu 3/4/21
19	Evaluate the website educational system through the result of test cases	4 days	Fri 3/5/21	Wed 3/10/21
20	Write the report about the problems need to be improved or solved	3 days	Thu 3/11/21	Mon 3/15/21
21	▪ To evaluation the entire website educational system	10 days	Tue 3/16/21	Mon 3/29/21
22	Create a survey to collect evaluation of user about the system	7 days	Tue 3/16/21	Wed 3/24/21
23	Evaluate the result of survey	3 days	Thu 3/25/21	Mon 3/29/21
24	▪ Conclusion	4 days	Tue 3/30/21	Fri 4/2/21
25	Evaluation the strength and weakness the entire system	2 days	Tue 3/30/21	Wed 3/31/21
26	Write the report about the plan for what need to improve in the future	2 days	Thu 4/1/21	Fri 4/2/21
27	▪ Write the final report	21 days	Fri 4/2/21	Fri 4/30/21
28	Write the final report	21 days	Fri 4/2/21	Fri 4/30/21

Appendix B – Final plan

Task Name	Duration	Start	Finish
Website Educational System Project			
To gather the requirements for the website educational system	194 days	Sat 8/8/20	Fri 4/30/21
Analysis the requirements	7 days	Sat 8/8/20	Mon 8/17/20
Write the report describe the specific functions for administrator, training department, teacher, student	5 days	Sat 8/8/20	Thu 8/13/20
	2 days	Fri 8/14/20	Mon 8/17/20
To analyze requirements and draw diagrams that represent the functions, operations, and relationships of requirements	16 days	Tue 8/18/20	Tue 9/8/20
Design the relationship between the necessary objects: Administrator, Training Department, Teacher, Category, Account, Course, Class ...	6 days	Tue 8/18/20	Tue 8/25/20
Design flow chart demonstrates the functions for each object: Administrator, Training Department, Teacher, Student	5 days	Wed 8/26/20	Tue 9/1/20
Design use case diagram demonstrates the activities for each object: Administrator, Training Department, Teacher, Student	5 days	Wed 9/2/20	Tue 9/8/20
To design the website educational system	32 days	Thu 9/10/20	Fri 10/23/20
Plan to design user interface for Administrator, Student, Training Department, Teacher	10 days	Thu 9/10/20	Wed 9/23/20
Plan to implement the functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for student	20 days	Thu 9/24/20	Wed 10/21/20
Choose methods, technologies using build the system	2 days	Thu 10/22/20	Fri 10/23/20
To implement the website educational system	90 days	Sat 10/24/20	Tue 2/23/21
Implement user interface for Administrator, Student, Training Department, Teacher pages	30 days	Sat 10/24/20	Tue 12/1/20
Design database for Administrator, Student, Teacher, Training Department, Course, Class, Category...	10 days	Wed 12/2/20	Tue 12/15/20
Implementation of functions: manage user accounts for admin, manage class schedules, organize subjects for the training department, manage students, create topics for each subject of the teacher, see the class schedule, upload report file for students...	50 days	Wed 12/16/20	Tue 2/23/21
To test the entire website educational system	14 days	Wed 2/24/21	Mon 3/15/21
Test the functions of website: add, update, delete account of administrator, add, update, delete category, course and arrange class schedules of training department...	7 days	Wed 2/24/21	Thu 3/4/21
Evaluate the website educational system through the result of test cases	4 days	Fri 3/5/21	Wed 3/10/21
Write the report about the problems need to be improved or solved	3 days	Thu 3/11/21	Mon 3/15/21
To evaluation the entire website educational system	10 days	Tue 3/16/21	Mon 3/29/21
Create a survey to collect evaluation of user about the system	7 days	Tue 3/16/21	Wed 3/24/21
Evaluate the result of survey	3 days	Thu 3/25/21	Mon 3/29/21

Conclusion	4 days	Tue 3/30/21	Fri 4/2/21
Evaluation the strength and weakness the entire system	2 days	Tue 3/30/21	Wed 3/31/21
Write the report about the plan for what need to improve in the future	2 days	Thu 4/1/21	Fri 4/2/21
Write the final report	18 days	Fri 4/2/21	Tue 4/27/21
Write the final report	18 days	Fri 4/2/21	Tue 4/27/21

Appendix C – Education System Functionals

EDUCATION SYSTEM				
NO	FUNCTION	ROLE	DESCRIPTION	SEND EMAIL TEMPLATE
1	AUTHENTICATION			
1.1	Login	Admin	As a User - Login to the Education System using my email and password -> I can be directed to the relevant home screen	
		Tutor		
		Student		
1.2	Forgot Password	Tutor	As a User - Reset my password via an email link if I forget my old password so that I can re-login with my email address and new password after I reset my password successfully	Reset password template
		Student		
2	MANAGE USER ACCOUNT			
2.1	Create Account	Admin	As an Administrator - Create the account for the users of the Education System - Send email for the users the email & password - Require user change password in the first-time login, when user	Notification creating account user template

			change password the status of this account will change from is_active: false -> is_active: true - Whenever user change the password the status just change - If the status of account: is_active: false, anytime user login -> redirect to change password screen	
2.2	List User Account		As an Administrator - See the list of all the student's account - See the list of all the tutor's account	
2.3	Manage User Account		As an Administrator - Delete user account	
3	CATEGORY			
3.1	Create Category	Admin	As an Administrator - Create the category	
3.2	Manage Category	Admin	As an Administrator - Edit category information - Delete category	
4	COURSE			

4.1	Create Course	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Send the request to create the new course (require administrator approval) - Set timescale - Automatically generate the secret key 	
		Student	<p>As a Student</p> <ul style="list-style-type: none"> - Receive the notification email when the new course is created 	Notification creating course template
4.2	Manage Course	Admin	<p>As an Administrator</p> <ul style="list-style-type: none"> - Can approve or reject the new course is created by the tutor 	
		Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Assign / Remove multi other tutors - Edit the course information - Delete the course 	
4.3	List Course	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - See the list of all the course along with the status: <ol style="list-style-type: none"> 1. Pending (waiting for administrator's approval) 2. On process (after approving) 3. Accomplish (after the end date of the course) 4. Reject (after teacher send request to create the course) 	

		Student	<p>As a Student</p> <ul style="list-style-type: none"> - See the list of all the course along with the status: <ol style="list-style-type: none"> 1. On process (after approving) 2. Accomplish (after the end date of the course) 	
4.4	Join Course	Student	<p>As a Student</p> <ul style="list-style-type: none"> - Click the button join course to send the request to join - Receive the secret key via email - Enter secret key to the input to join the course 	Receive secret key join course template
4.5	Leave Course	Student	<p>As a Student</p> <ul style="list-style-type: none"> - Leave the course, when the student leaves the course, the student will fill the reason at input reason and alert the user can't join this course again (create a record save courseId, studentId, reason) - If a student leaves the course, the student can't join the course again. (Check if the record, the student can't join) - Student can join the course again if the tutor removes this student from the student leave course list 	
4.6	List Student Leave Course	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - See the list of all student leave the course with the reason 	
4.7	Manage Student Leave Course	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Delete the student from the list of students leaving the course 	

4.8	Upload Course Video	Tutor	As a Tutor - Upload course videos	
4.9	List Course Video	Tutor	As a User - See the list of all the course videos	
		Student	- Support view course videos in the browser	
4.10	Manage Course Video	Tutor	As a Tutor - Edit course videos information - Delete course videos	
5	BLOG			
5.1	Create Blog	Tutor	As a Tutor - Create a new blog in the course - All user in the course will receive the notification email whenever tutor or student create a blog	Notification creating blog template
		Student	As a Student - Send request to create blog	
5.2	Upload File	Tutor	As a User - Upload file with the blog	
		Student		

5.3	List All Blog	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - See the list of all the blog's student along with the status: 1. Pending 2. Approved 3. Rejected 	
		Student	<p>As a Student</p> <ul style="list-style-type: none"> - See list of all the approved blogs in the course 	
5.4	Download File	Tutor	<p>As a User</p> <ul style="list-style-type: none"> - Download file 	
		Student		
5.5	List Own User Blog	Tutor	<p>As a User</p> <ul style="list-style-type: none"> - See the list of all own user's blog 	
		Student		
5.6	Approve / Reject Blog	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Approve or reject the student's blog 	
5.7	Manage Blog	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Edit own blog information - Can delete own blog and all student's blog 	
		Student	<p>As a Student</p> <ul style="list-style-type: none"> - Edit own blog information - Delete own blog 	
5.8	Like Blog	Tutor		

		Student	As a User - Like user's blog	
5.9	Like Comment	Tutor	As a User	
		Student	- Like user's comment	
5.10	Comment Blog	Tutor	As a User	
		Student	- Comment on user's blog	
6	ACTIVITY			
6.1	Create Activity	Tutor	As a Tutor - Create the new activity in the course - Set deadline	
6.2	List Activity	Tutor	As a Tutor - See list of all the activity in the course	
		Student	As a Student - See list of all the activity in the course	
6.3	List Assignment / Coursework	Tutor	As A Tutor - See the list of all the assignment/coursework of the student in each activity	

6.4	Manage Activity	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Edit activity information (edit deadline, limit the number of file upload...) along with the status: <ol style="list-style-type: none"> 1. On process 2. Overdate 3. Hidden (Can't submit by the changing status of tutor) 	
6.5	Upload Assignment / Coursework	Student	<p>As a Student</p> <ul style="list-style-type: none"> - Upload the assignment/coursework file includes Word file, PDF file, Image... 	
6.6	Upload Requirement	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Upload the requirement file includes Word file, PDF file... - When student uploads, the tutor will get the notification via email 	Notification upload requirement/ create activity template
6.7	Download File	Tutor	<p>As a User</p> <ul style="list-style-type: none"> - Download file 	
6.8		Student	<p>As a Student</p> <ul style="list-style-type: none"> - Comment on each assignment/coursework in each activity 	
	Comment Activity	Tutor	<p>As a Tutor</p> <ul style="list-style-type: none"> - Comment on each assignment/coursework of each student in the activity 	
		Student		

6.9	Grade Activity	Tutor	As a Tutor - Grade for all activities of each student	
6.10	View Grade	Student	As a Student - See the grade of each activity	
7	CHAT			
7.1	List Contact	Student	As a User - See list of all the user	
		Tutor		
7.2	Search Contact	Tutor	As a User - Search contact	
		Student		
7.3	Create Message	Tutor	As a User - Create message (send message)	
		Student		
7.4	Delete Message	Tutor	As a User - Delete message	
		Student		
8	REPORT SYSTEM			
8.1	Report	System	The system automatic generate the report statistic - The course is cancelled	

			- The course is accomplished - The course is rejected	
8.2	Export Report PDF	Tutor Admin	As a Tutor - Export PDF the report statistic	

Appendix D – Education System Screenshots

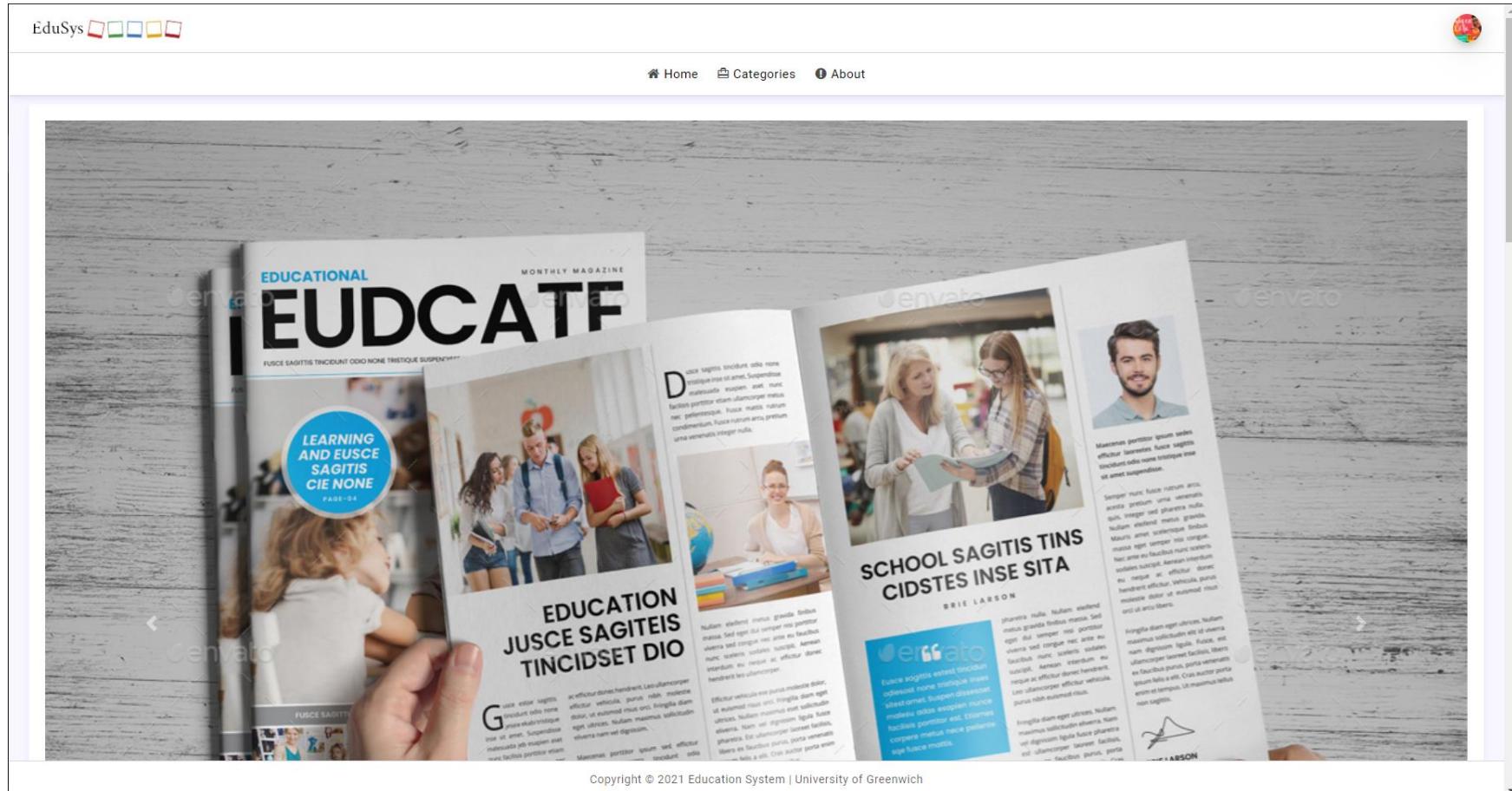


Figure 119 Home page

EduSys 

Home Categories About

Courses

[NEW COURSE](#)

Choose courses All



C Programming
Introductory C Programming
On process
Doan Trung Tung  

Beginning C++ Programming - From Beginner to Beyond
On process
Doan Trung Tung  

Learn and Understand NodeJS
On process
Doan Trung Tung  



TUTORIAL
HTML CSS Generics 

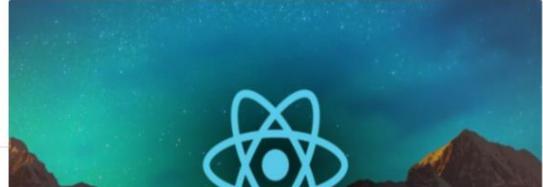


Figure 120 List course page – tutors

New course X

TITLE *

DESCRIPTION *

START DATE * END DATE *

 □ □

BACKGROUND IMAGE

No file chosen



Default background image

CREATE

Figure 121 Create new course form

New course X

TITLE *

DESCRIPTION *

START DATE * CALENDAR **END DATE *** CALENDAR

BACKGROUND IMAGE
 No file chosen

Default background image

UPDATE

Figure 122 Edit course information form

EduSys

Home Categories About

Course Detail
Categories / Programming / Introductory C Programming

Introductory C Programming

Doan Trung Tung

Duration: 17-Apr-2021 - 21-Apr-2021

Category: Programming

Total students: 1

COURSE BLOG ACTIVITY

Description

Learn Essential Programming Fundamentals. Master programming skills to solve complex problems.



Copyright © 2021 Education System | University of Greenwich

Figure 123 Course detail page – tutors



Figure 124 Blog page – tutors

Blog detail

X



Pham Thi Ngoc



Introduce the chart package

4 days ago

In order to build a chart, you can go to react-chartjs-2 at the below link
<https://www.npmjs.com/package/react-chartjs-2>



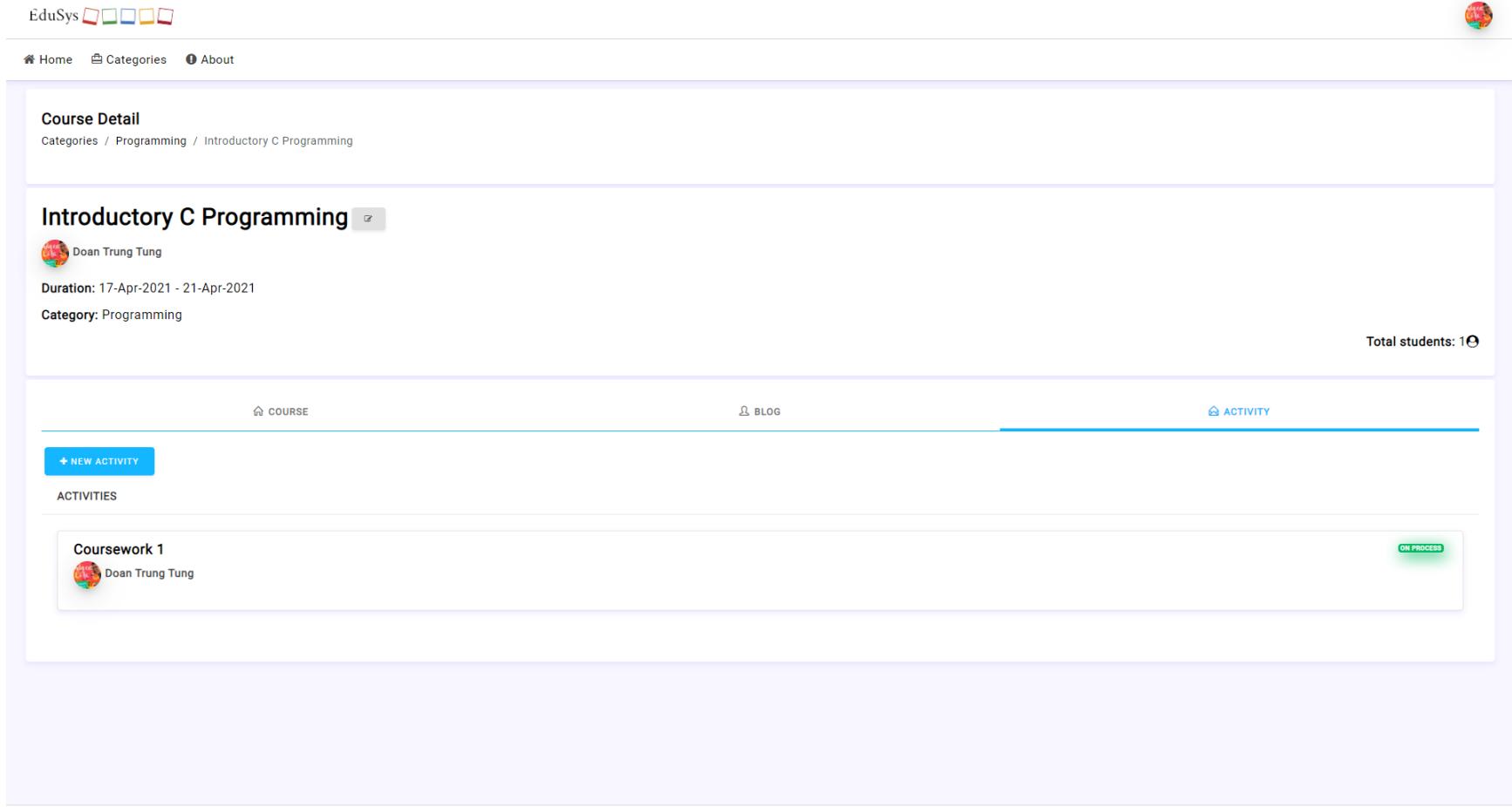
Doan Trung Tung

Thanks for sharing!!!

4 days ago

Write comment here...

Figure 125 Blog detail



EduSys 

Home Categories About

Course Detail
Categories / Programming / Introductory C Programming

Introductory C Programming 

 Doan Trung Tung

Duration: 17-Apr-2021 - 21-Apr-2021

Category: Programming

Total students: 1 

 COURSE  BLOG  ACTIVITY

+ NEW ACTIVITY

ACTIVITIES

Coursework 1  Doan Trung Tung 

Copyright © 2021 Education System | University of Greenwich

Figure 126 Activity page

Activity Detail

x

COURSEWORK 1 ON PROCESS

CREATED BY  Doan Trung Tung

DESCRIPTION Limit 5000 words

DEADLINE 17-Apr-2021 - 20-Apr-2021

REQUIREMENTS  coursework.docx 

ATTACH FILE No file chosen

FILE SUBMISSIONS View all submissions

Figure 127 Activity detail

All submissions x

 Pham Thi Ngoc PASS ▾

 Conceptual Design Assignment.pdf 

Figure 128 List student's submission and grade

Appendix E – Test Case

No	Function	Role	Action/Data	Expected result	Actual result	Status
1	<i>Authentication</i>					
1.1	Login	Admin	Enter valid username and password	Login successful and browser redirect to home page	Login successful and browser redirect to home page	Pass
			Enter invalid username or password	Login failed	Login failed and browser display warning message	Pass
		Student	Leave username or password blank	Login failed	Login failed and browser display warning message	Pass
1.2	Forgot password	Tutor	Enter valid email	Receive the email reset password	Browser display success message	Pass
			Enter invalid email	Display warning message	Display warning message	Failed
2	<i>User account</i>					
2.1	Create new account	Admin	Enter valid information	Create new account success	Create new account success	Pass
			Enter invalid information	Create new account failed	Browser display warning message	Pass

2.2	List user accounts	Admin	Click user tab	Display list of all the user information	Display list of all the user information	Pass
2.3	Manage user account	Admin	Update information with valid information	Update user account information success	Update user account information success	Pass
			Delete the user	Delete successful	Delete successful	Pass
3	Category					
3.1	Create new category	Admin	Enter valid information	Create new category successful	Create new category successful	Pass
3.2	Manage categories	Admin	Edit category information with valid information	Edit category information successful	Edit category information successful	Pass
			Delete the category	Delete category successful	Delete category successful	Pass
4	Course					
4.1	Create course	Tutor	Enter valid information	Create course successfull	Create course successfull	Pass
4.2	Manage course	Admin	Approve or Reject the new course	Approve or Reject successful	Approve or Reject successful	Pass
		Student		Receive the email notify the new course is created	Receive the email notify the new course is created	Pass
4.3	List course	Tutor			See the list of all the course by category	Pass

		Student	Hover category tab and choose one of them	See the list of all the course by category		
4.4	Join Course	Student	Click choose course which not join	The modal display to enter secret key	The modal display to enter secret key	Pass
4.5	Upload video in course	Tutor	Click button upload video button, enter and choose valid information	Upload video successfull	Upload video successfull	Pass
5	<i>Blog</i>					
5.1	Create blog	Tutor	Enter valid information	Create blog successfull	Create blog successfull	Pass
		Student				
5.2	Manage student's blog	Tutor	Approve or reject student's blog	Approve or reject successful	Approve or reject successful	Pass
5.3	Upload file	Tutor	Choose the file upload with blog	Upload successful	Upload successful	Pass
		Student				
5.4	List all blogs	Tutor	Click blog tab	See the list of all the blogs	See the list of all the blogs	Pass
		Student				
5.5	List own blog	Tutor	Click personal blog tab	See the list of all own blog	See the list of all own blog	Pass
		Student				

5.6	Manage own blog	Tutor Student	Edit blog information	Edit blog information successfull	Edit blog information successfull	Pass
			Delete the blog	Delete blog successfull	Delete blog successfull	Pass
5.7	Comment on blog	Tutor	Write content and press enter	The comment appears on blog	The comment appears on blog	Pass
		Student				
6	Activity					
6.1	Create activity	Tutor	Enter valid information	Create activity successfull	Create activity successfull	Pass
6.2	List acitivy	Tutor	Click activity tab	Show list of all the acities	Show list of all the acities	Pass
		Student				
6.3	List assignment/coursework files	Tutor	Click view activity detail and click view all submissions	Show list of all the assignment/coursework files	Show list of all the assignment/coursework files	Pass
		Student				
6.4	Manage acitivy	Tutor	Edit activity information	Edit activity information successful	Edit activity information successful	Pass
6.5	Upload requirement files	Tutor	Choose the file and click upload button	Upload successful	Upload successful	Pass
	Upload assignment/coursework files	Student				

6.6	Download file	Tutor Student	Click the file	Opening save file window as a ZIP file	Opening save file window as a ZIP file	Pass
6.7	Grade activity	Tutor	Click dropdown menu and choose the grade	Grade successfull	Grade successfull	Pass
6.8	View grade	Student	Click view activity detail	See the grade	See the grade	Pass
7	<i>Chat</i>					
7.1	List contact	Tutor Student	Click chat page	See the list friend sent a message	See the list friend sent a message	Pass
7.2	Search contact	Tutor Student	Type search contact by name	Display the contact is match with typing	Display the contact is match with typing	Pass
7.3	Send messages	Tutor Student	Send messages, images or videos	Send message successfull	Send message successfull	Pass
8	<i>Report System</i>					
8.1	View report statistic	Admin	Click dashboard page	Display the chart statistic	Display the chart statistic	Pass

Appendix F – Wireframes

The wireframes of the Education System detail at the below link

<https://drive.google.com/file/d/1kPJ8HPiv52dHpb-iZrYq74wt3jFcwpdx/view?usp=sharing>