

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



WEB PROGRAMMING (CO3049) - CC01

Assignment Report

Multiple-Choice Quiz Website

Advisor: PhD. Nguyễn Đức Thái

Class: CC01

Students: Nguyễn Đình Sáng - 1952955
Trần Nhật Tân - 1852732
Nguyễn Thụy Khánh Linh - 1952319
Võ Phạm Hoài Nam - 1952863
Võ Trần Minh Đạt - 1913086

HO CHI MINH CITY, NOV 2021



Contents

1 Member list & Workload	3
2 Requirement Gathering and Analysis	4
2.1 Requirements Gathering	4
2.2 Stakeholders	4
2.3 Software Requirement Specification (SRS)	4
2.3.1 Purpose	4
2.3.2 Project Scope	4
2.3.3 Overall Description	4
2.3.4 Target Audience	5
2.4 Functional Requirements	5
2.4.1 Teacher side	5
2.4.2 Student side	6
2.5 Non-functional Requirements	6
2.6 Use-case Diagram	7
2.6.1 Class Diagram	10
2.6.2 Activity Diagram	11
3 Design	12
3.1 Architecture	12
3.2 Front End: Interface and Interactions	13
3.3 Back End: Database	15
3.3.1 Why we choose MongoDB?	15
3.3.2 Database implementation	15
4 Implementation	18
4.1 Teacher Implementation	18
4.1.1 Course Page	18
4.1.2 Quiz Page	19
4.1.3 Question Page	21
4.1.4 Result Page	23
4.1.5 Responsive Pages	24
4.1.5.a Course Page	24
4.1.5.b Quiz Page	25
4.1.5.c Question Page	27
4.1.5.d Result Page	29
4.2 Student Implementation	31
4.2.1 Home Page	31
4.2.2 Quiz Page	33
4.2.3 View score Page	35
4.2.4 Responsive Pages	36
4.2.4.a Home Page	36
4.2.4.b Result Page	37



5 Testing	38
5.1 What is Unit Testing and Integration Testing?	38
5.2 Teacher Testing	39
5.2.1 Unit Test	39
5.2.2 Integration Test	44
5.3 Student Testing	49
5.3.1 Unit Test	49
5.3.2 Result of Unit Test	50
6 Evaluation	51
6.1 Search Engine Optimization (SEO)	51
6.1.1 What is SEO?	51
6.1.2 Why SEO is important	51
6.1.3 What we use to improve SEO.	52
6.2 Website Rating Point	53
7 Conclusion	54
7.1 Source code and domain	54
7.2 Results review	54
7.3 What we learned?	54



1 Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Nguyễn Đình Sáng	1952955	Design and testing	100%
2	Trần Nhật Tân	1852732	Back-end and testing	100%
3	Nguyễn Thúy Khánh Linh	1952319	Front-end and SEO	100%
4	Võ Phạm Hoài Nam	1952863	Back-end and testing	100%
5	Võ Trần Minh Đạt	193086	Front-end and SEO	100%



2 Requirement Gathering and Analysis

2.1 Requirements Gathering

Since science and technology have advanced significantly, eLearning is being used in schools at all levels, as well as education and training centers. Online quizzes have become one of the most common eLearning patterns for many online courses. They are used to educate and evaluate students.

Online quiz system is a web application that establishes a network between the lecturers and the students. Lecturers login to the site then create courses and quizzes with many questions. These questions are displayed as a test to the students. The answers enter by the students are then evaluated and their score is calculated and saved. This score then can be accessed by the lecturers to evaluate students' performance.

2.2 Stakeholders

Stakeholders	Responsibilities
Lecturer	<ul style="list-style-type: none">– Create and manage courses that they are responsible for teaching.– Create and edit (add, remove questions,...) quizzes for each course.– Manage the information and scores of students who have completed their quizzes.
Student	<ul style="list-style-type: none">– Take the quizzes before deadline and after opening time.– Manage the score of each of their finished quizzes.

2.3 Software Requirement Specification (SRS)

2.3.1 Purpose

The QUIZ - multiple-choice quiz website, which is developed by our team, will be undertaken mainly by high schools, higher-education schools and training centers in order to support eLearning system, educate and evaluate students, also keep track of the progress and quality of student learning. Besides, online quiz system can also improve most of the disadvantages of paper tests.

2.3.2 Project Scope

The Online quiz system allows users to sign in using one of two types of accounts: *Teacher* or *Student*. Different privileges and functions are assigned to each type of account. Functions of Teacher accounts include course creation, quiz management and attendee management. Student accounts will have quiz taking and score management.

Unlike a paper test, our QUIZ website provides additional support to creators and participants in terms of score management and attendee management. This online quiz website reduces the manual work, maintains accuracy, increases efficiency and saves time.

2.3.3 Overall Description

In QUIZ , lecturer can login to create courses. For each course, they can add a variety of quizzes with varying content, knowledge, and purposes. Following that, they insert multiple-choice questions into each quiz and review the results later. Students can take the quizzes that are currently displayed on their website, or they can take another quiz by searching for or entering the correct quiz's ID. Students may take a quiz more than once as long as they do so before the deadline.



The system compares their answers to the correct answer marked by the lecturer and computes the grade based on the correct answers and question levels. The results will be shown to the students on the screen.

2.3.4 Target Audience

- **Basic English level:** Because we construct our system in English, users of our QUIZ website must have a basic understanding of English in order to utilize it efficiently.
- **Basic knowledge of Technology:** To use and maintain the system efficiently, users must have a basic understanding of technology.
- **High school and Higher Education environment:** Our system can be used for many levels of education, but we focus primarily on high school and higher education students because they are mature enough and have enough knowledge to use the system effectively. Also, eLearning is much more widely and efficiently used in High school and Higher Education environment than other lower academic levels.

2.4 Functional Requirements

2.4.1 Teacher side

Function	Description
Authenticate Account	Lecturers can sign in to their accounts and gain the privileges of managing their students, courses and quizzes on the website.
Manage Courses	<ul style="list-style-type: none">– Lecturers can create courses that they are responsible for teaching.– Lecturers can also edit or delete their created courses.
Manage Quizzes	<ul style="list-style-type: none">– Lecturers can create quizzes for each of their course by adding questions with its 4 options respectively.– Lecturers can also delete their created quizzes.– Lecturers can also edit their quizzes by edit the questions.
Manage Quiz Questions	<ul style="list-style-type: none">– Lecturers can add many questions for a new quiz.– For existing quizzes, lecturers can add more questions, edit the existing questions or delete questions.
Download and Print PDF	Lecturers can download and print a PDF file containing the question of a specific quiz.
View Students' results	For each of their quiz, Lecturers can view the final results of the students who have finished it.



2.4.2 Student side

Function	Description
Create and Authenticate Account	Students can register or sign in to their existing accounts and gain the privileges of managing their taken quizzes and scores.
Do Quizzes	– Students can do quizzes displayed on the "Your Quizzes" page. – Students can take their desired quiz by entering the quiz ID.
Review Quizzes	Students can review their completed quizzes by going to the "View score" page. For each completed quiz, they can see the final score, all of the questions with their correct answers, and whether they did it correctly.
Search Quizzes	Students can go to the Search page and find all of the quizzes that relevant to the keyword they entered.

2.5 Non-functional Requirements

– Availability:

User can access to the Website 24/7. Students can take the quiz only after the opening time and before the deadline, however can open site anytime to access other information. Lecturers can create courses and quizzes anytime.

– Responsive Website:

Users are able to use our website in variety of platforms and devices. It will offers an optimized browsing experience for users.

– Usability:

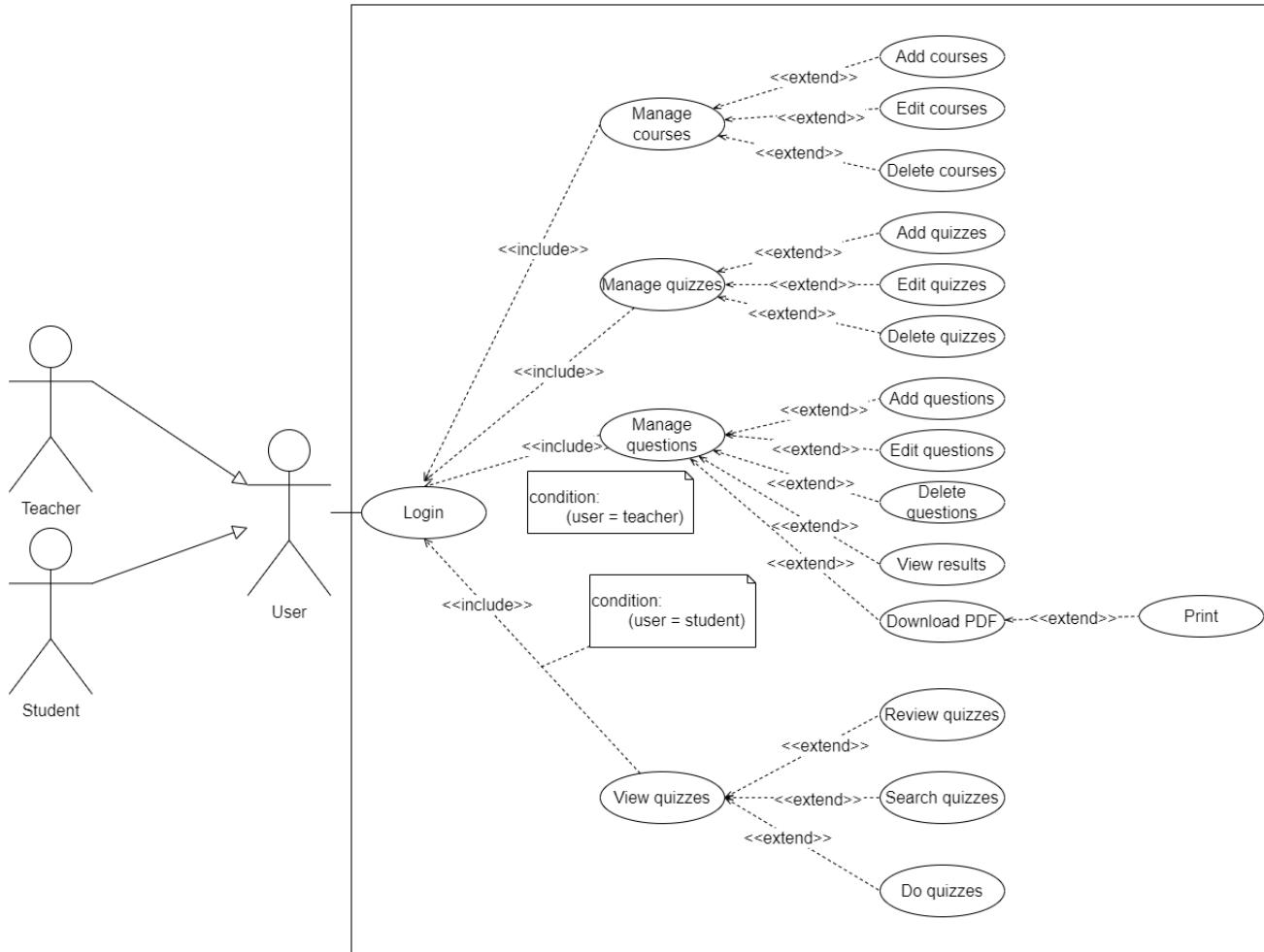
User needs to have an account to login to the website. The server should have well session management to maintain sessions in the application.

– Reliability:

Students must not be able to view the quiz before its opening day and do the quiz after its deadline. Besides, a user must login to use this system and prevented from trying to access to another pages of the website.

2.6 Use-case Diagram

QUIZ



Use case diagram

Description

- Manage courses:



Use case name	Manage Course
Actor	Teacher
Description	Teacher can manage course (insert, edit or remove).
Pre-condition	1. Must have teacher account
Normal flow	1. The system displays courses. 2. Teacher can view the quizzes in course. 3. End of use case.
Alternative flow	Alternative 1: At step 2. 2a. If Teacher add course, the system shows the form for teacher to fill in. 2b. If Teacher edit a specific course, the system shows the form for teacher to fill in. 2c. If Teacher delete course, the system warns teacher and let teacher to confirm again.
Exceptions	Exception 1: At step 2a, 2b, and 2c. If Teacher cancels, the system hides the form. Exception 2: at step 2a, 2b. If Teacher did not fill all the information of the course, the system will prevent Teacher from submitting.

- Manage quizzes:

Use case name	Manage Quiz
Actor	Teacher
Description	Teacher can manage quiz (insert, edit or remove).
Pre-condition	1. Must have a course.
Normal flow	1. The system displays quizzes. 2. Teacher can view the questions in quiz. 3. End of use case.
Alternative flow	Alternative 1: At step 2. 2a. If Teacher add quiz, the system shows the form for teacher to fill in. 2b. If Teacher edit a specific quiz, the system shows the form for teacher to fill in. 2c. If Teacher delete quiz, the system warns teacher and let teacher to confirm again.
Exceptions	Exception 1: At step 2a, 2b, and 2c. If Teacher cancel, the system hides the form. Exception 2: at step 2a, 2b. If Teacher did not fill all the information of the quiz, the system will prevent Teacher from submitting.

- Manage questions:

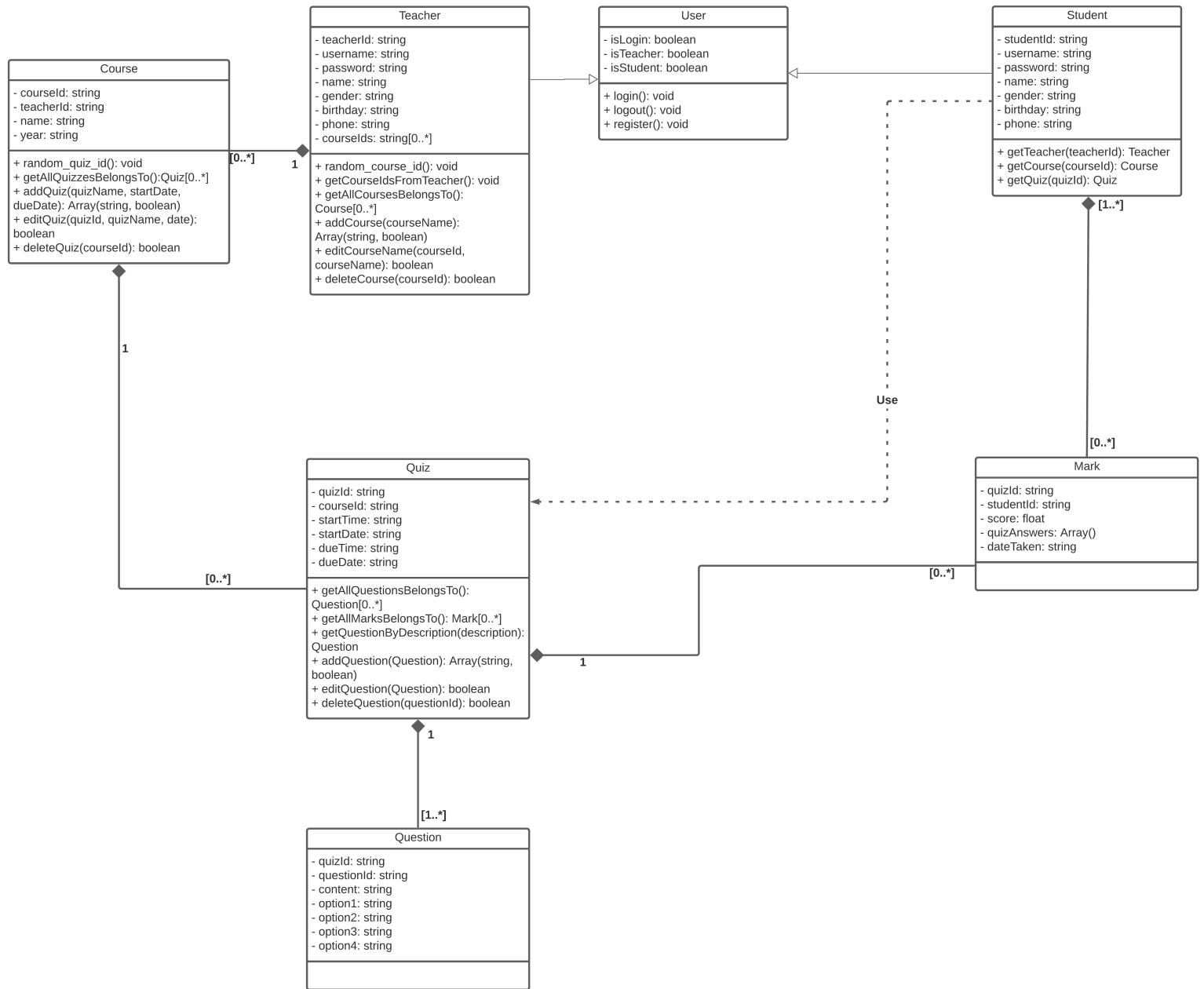


Use case name	Manage Question
Actor	Teacher
Description	Teacher can manage question (insert, edit or remove). Teacher can view student's scores.
Pre-condition	1. Must have a quiz.
Normal flow	1. The system displays questions. 2. Teacher can manage the questions in quiz. 3. Teacher can view scores of students. 4. End of use case.
Alternative flow	Alternative 1: At step 2. 2a. If Teacher add quiz, the system shows the form for teacher to fill in. 2b. If Teacher edit a specific quiz, the system shows the form for teacher to fill in. 2c. If Teacher delete quiz, the system warns teacher and let teacher to confirm again. Alternative 2: At step 3. 3a. Teacher can download their questions as PDF file.
Exceptions	Exception 1: At step 2a, 2b, and 2c. If Teacher cancel, the system hides the form. Exception 2: at step 2a, 2b. If Teacher did not fill all the information of the question, the system will prevent Teacher from submitting.

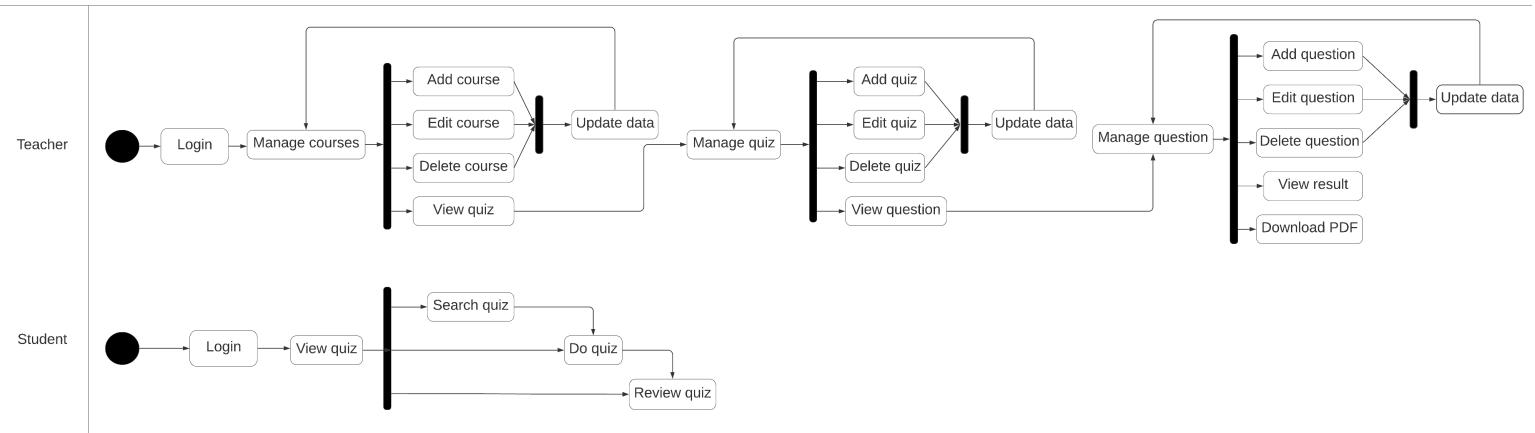
- View quizzes:

Use case name	View Quizzes
Actor	Student
Description	Student can view their recently done quiz. Student can search quiz by ID or name. Student can do the quiz and review their score.
Pre-condition	Must have student account.
Normal flow	1. The system display recently done quizzes. 2. Student can do quiz by quizID given by Teacher. 3. Student review their work. 4. End of use case.
Alternative flow	Alternative 1: At step 2. 2a. Student can search quiz by name and then do the quiz.
Exceptions	Exception 1: At step 2. 2a. Student can skip step 2 and go to step 3.

2.6.1 Class Diagram



2.6.2 Activity Diagram

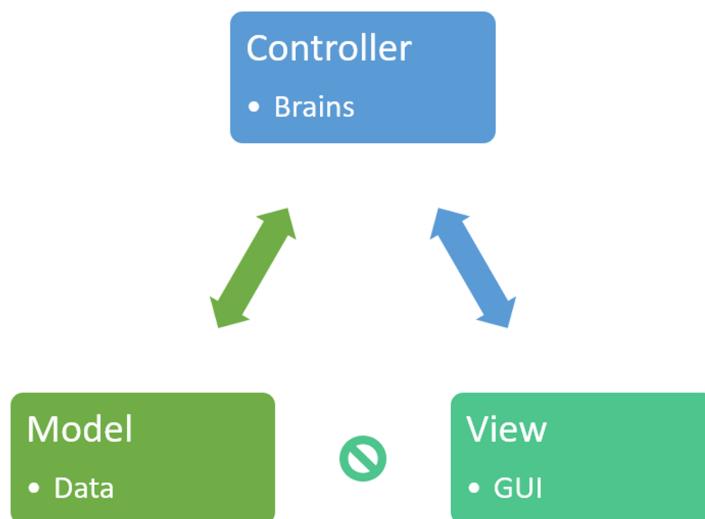


3 Design

3.1 Architecture

MVC is popular in app and web development, and it's one of the most widely used software design patterns for app and web development. The Model View Controller design pattern separates concerns into one of 3 buckets:

- Model
- View
- Controller



MVC diagram

Model: stores & manages data. Often a database, in our quick example we'll use local web storage on a browser to illustrate the concept.

View: Graphical User Interface The view is a visual representation of the data- like a chart, diagram, table, form.

The view contains all functionality that directly interacts with the user - like clicking a button, or an enter event.

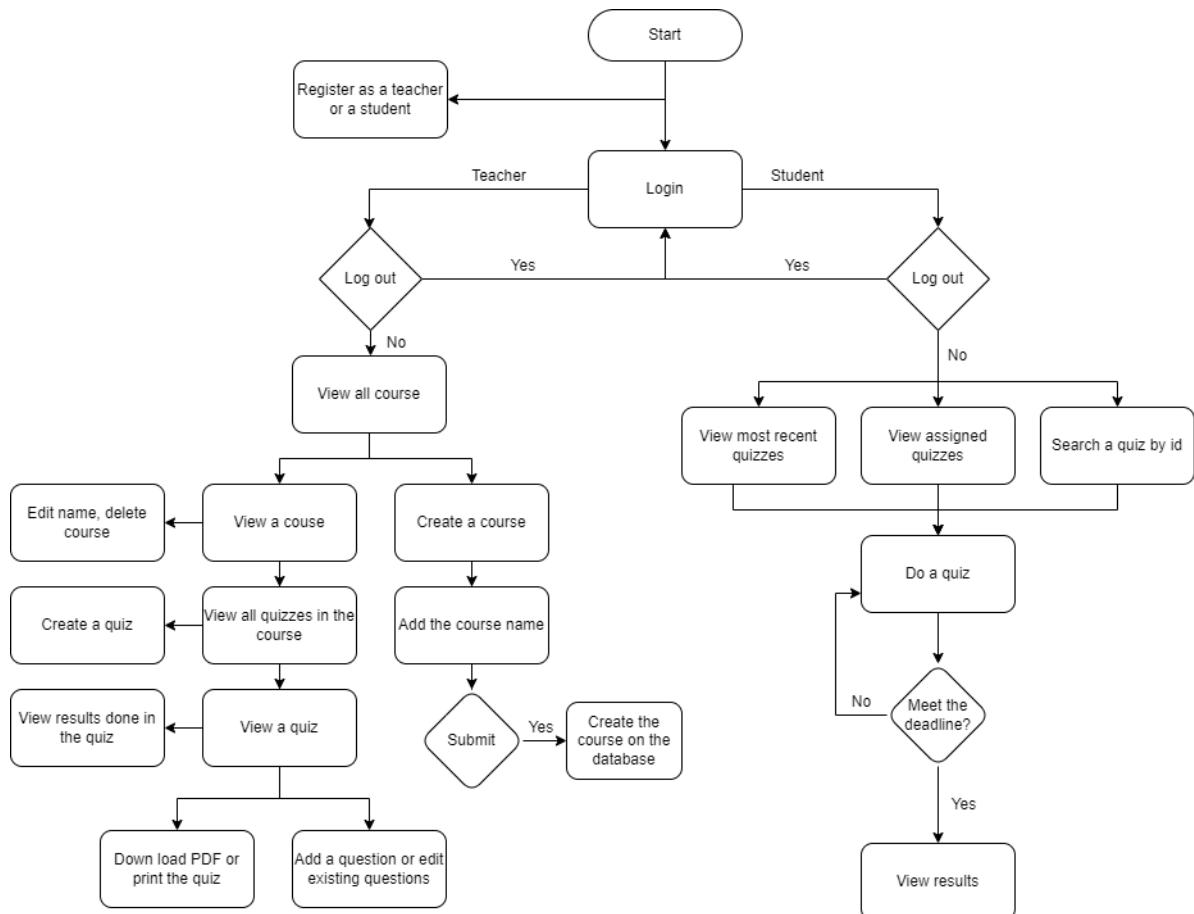
Controller: Brains of the application. The controller connects the model and view. The controller converts inputs from the view to demands to retrieve/update data in the model.

The controller receives input from view, uses logic to translate the input to a demand for the model, the model grabs the data, the controller passes data from the model back to the view for the user to see in a nice display.

3.2 Front End: Interface and Interactions

There are many different pathways a user can take when interacting with a product. A user flow is a visual representation, either written out or made digitally, of the many avenues that can be taken when using an app or website. The flowchart begins with the consumer's entry point on the product, like an onboarding screen or homepage, and ends with the final action or outcome, like purchasing a product or signing up for an account. Depicting this process allows designers to evaluate and optimize the user experience and therefore increase client conversion rates.

In this web, we drew a user flow diagram to efficiently design a mockup:



User flow on multiple-choice quiz website



This is overview of the web mockup. See details in the section [Implementation]

The teacher mockup design includes three main pages:

- Admin home page:** Shows a list of "All courses" with icons for "Computer networks", "Programming integration project", and "Web Programming". Each course has a "View" button.
- Course page:** Shows a "Course page" for "Web Programming" with a "Midterm exam" section. It displays the exam period (2020-12-02 to 2020-12-08) and a "View" button.
- Result page:** Shows a "Result page" for the "Midterm exam" with columns for "No.", "Name", and "Score".

Overview of teacher mockup design

The student mockup design includes five main pages:

- Student HomePage:** Features a "Let's challenge yourself with quizzes!" section and a "HAPPY QUIZZES" section showing "Most Recent Quizzes".
- Your Quizzes:** Shows a list of quizzes with columns for "Quiz Name", "Description", and "Start Date".
- Quiz-Selection:** A purple screen with a "QUIZ ITU" logo and a "QUIZ" button.
- Quiz Begin:** A blue screen asking for "QUIZ NAME" and featuring a "Start Quiz!" button.
- Question Screen:** A blue screen displaying a question: "2. Maximum number of nodes in a binary tree with height h, where root's height 0, is" with options: A) 2^h, B) 2^(h+1), C) 2^h-1, D) 2^(h+1)-1, and a "Next" button.

Overview of student mockup design



3.3 Back End: Database

3.3.1 Why we choose MongoDB?

MongoDB Atlas is a fully-managed database as a service that runs on all public clouds. It scales from a free tier (no credit card required) to global clusters.

Using MongoDB can provide many benefits to a software development team. Its flexible schema makes it easy to evolve and store data in a way that is easy for programmers to work with. MongoDB is also built to scale up quickly and supports all the main features of modern databases such as transactions. Additionally, MongoDB has a large community of users that can provide help, and enterprise-level support is available.

3.3.2 Database implementation

In our database we implemented 6 collections: Teacher, Student, Course, Quiz, Question, and Mark. A document of each collection is presented as follow:

Teacher	Student	Course	Quiz	Question	Mark
teacherId: String	studentId: String	courseld: String	quizId: String	description: String	studentId: String
username: String	username: String	teacherId: String	name: String	level: Int32	score: Double
password: md5 String	password: md5 String	name: String	startDate: String	unitScore: Double	totalScore: Double
name: String	name: String	year: String	dueDate: String	option1: String	quizAnswer: Array[String]
gender: String	gender: String		teacherId: String	option2: String	quizId: String
birthday: String	birthday: String		courseld: String	option3: String	dateTaken: String
phone: String	phone: String			option4: String	
courselds: Array[String]				quizId: String	
				courseld: String	

Database schema implementation

The below pictures show document records from each of the collections.

```
_id: ObjectId("618de3d38e3e0000ac0072d3")
teacherId: "TC-4203517869"
username: "tc.wZAt@gmail.com"
password: "2a8098c5d332f45b6893f71f961dc3a6"
name: "Nguyen Minh Hoi"
gender: "Male"
birthday: "25/07/1972"
phone: "0935146702"
courseIds: Array
  0: "C-3759642018"
```

A document record from Teacher collection



```
_id: ObjectId("618deb728e3e0000ac0072ed")
studentId: "STU-0195872364"
username: "stu.Gciv@gmail.com"
password: "57d5a92aec4a83682b7974fdb1c59d22"
name: "Le Minh Nao"
gender: "Male"
birthday: "08/01/1960"
phone: "0421689730"
```

A document record from Student collection

```
_id: ObjectId("6197ad21180900006d002d02")
courseId: "C-7960351842"
teacherId: "TC-4203517869"
name: "Data structures and algorithms"
year: "2020"
```

A document record from Course collection

```
_id: ObjectId("6197ae8d180900006d002d05")
quizId: "Q-4238061597"
name: "Midterm exam"
startDate: "2021-01-10"
dueDate: "2021-12-14"
teacherId: "TC-4203517869"
courseId: "C-7960351842"
```

A document record from quiz collection

```
_id: ObjectId("6197b27d180900006d002d09")
description: "For a binary search algorithm to work, it is necessary that the array ..."
level: 1
unitScore: 10
option1: "sorted"
option2: "unsorted"
option3: "in a map"
option4: "in a stack"
quizId: "Q-4238061597"
courseId: "C-7960351842"
```

A document record from question collection



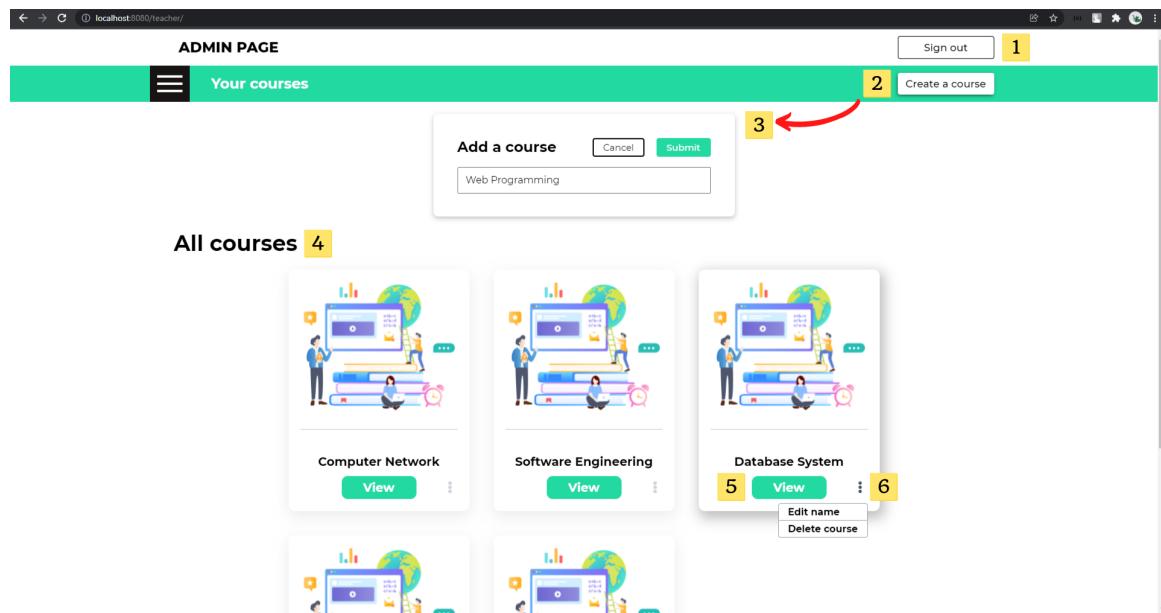
```
_id: ObjectId("6197b6ad957500005a00436c")
studentId: "STU-9101587797"
score: 20
totalScore: 90
quizAnswer: Object
  6197b27d180900006d0... : "true"
  6197b27d180900006d0... : "true"
  6197b27d180900006d0... : "false"
  6197b27d180900006d0... : "false"
  6197b27d180900006d0... : "false"
quizId: "Q-4238061597"
dateTaken: "2021-11-19 21:37:33"
```

A document record from Mark collection

4 Implementation

4.1 Teacher Implementation

4.1.1 Course Page

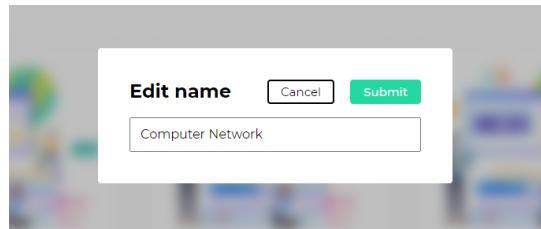


Hình 1: Course Page

No	Label	Annotation
1	Sign out	Clicking on the button, Firstly, the system will navigate the user to login page. Secondly, the system will clear the user session.
2	Create a course	Clicking on the button, the system will show the add course form.
3	Add a course form	User enter the name for new course, then (1) Click the Submit button to confirm creating new course. (2) Click the Cancel button to cancel creating new course.
4	All courses	Show all courses belongs to the teachers.
5	View	Clicking on the button, the system will navigate user to the Quiz page, which contains all quizzes belongs to the chosen course.
6	Interaction menu	Clicking on the button, the system will show two interaction options: (1) Edit: to edit the course information (name). (2) Delete: to delete the course.

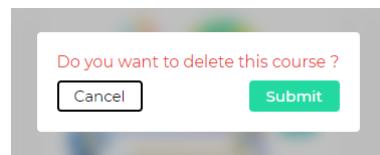


When the user selects the editing option, the editing course form will pop up in the center of the screen. We have handled the situation that input field of edited course name is empty, so the user need to fill in the course name if they want to update.



Hình 2: Edit Course Form

When the user selects the delete option, the delete course form will pop up in the center of the screen.



Hình 3: Delete Course Pop-up

4.1.2 Quiz Page

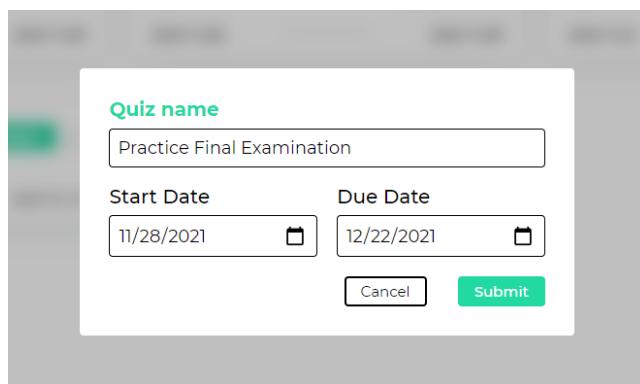
The screenshot shows the 'ADMIN PAGE' interface. At the top, there's a navigation bar with 'Your courses > Web Programming'. On the right, there are 'Sign out' and 'Add a quiz' buttons. A red arrow points from a yellow box labeled '1' to the 'Add a quiz' button. Below the navigation, a form for creating a new quiz is displayed. It includes fields for 'Quiz name' (labeled '2') containing 'Practice Final Examination', 'Start Date' (11/28/2021), 'Due Date' (12/22/2021), and difficulty levels ('Easy'). There are two questions listed under 'Question 1': 'What is the use of `isset()` function in PHP?' (labeled '3') and 'Option A*' (labeled '4'). Another question 'Option C' is also shown. At the bottom of the quiz creation form are 'Cancel' and 'Submit' buttons, with a red arrow pointing from a yellow box labeled '5' to the 'Submit' button. Below the quiz creation form, there's a section titled 'All quizzes' with three entries: 'Quiz 1' (2021-11-28), 'Quiz 2' (2021-11-29), and 'Quiz 3' (2021-11-21). Each quiz entry has a 'VIEW' button and a more options menu. A red arrow points from a yellow box labeled '6' to the 'VIEW' button of Quiz 3. Another red arrow points from a yellow box labeled '7' to the more options menu of Quiz 3, which shows 'Edit quiz' and 'Delete quiz' options.

Hình 4: Quiz Page



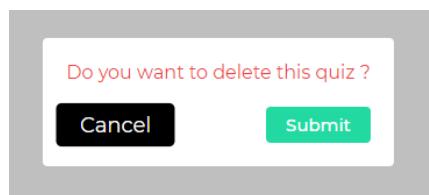
No	Label	Annotation
1	Add a quiz	Clicking on the button, the system will show the add quiz form.
2	Add quiz form	User enter the all the information of new quiz included (1) Quiz's name. (2) Start date, due date. (3) Information of questions of the quiz. Note that: In this form, we specify that Question A is the correct answer in default.
3	Add a question	Clicking on the button, the "Add quiz" form will extend to the bottom, in order to make place for new question.
4	Submit	Clicking on the button, confirm creating new quiz.
5	Cancel	Clicking on the button, cancel creating new quiz and the system will close the form.
6	View	Clicking on the button, the system will navigate user to the Question page, which contains all questions belongs to the chosen quiz.
7	Interaction menu	Clicking on the button, the system will show two interaction options: (1) Edit: to edit the quiz information (name, start date, due date) (2) Delete: to delete the quiz.

When the user selects the editing option, the editing quiz form will pop up in the center of the screen. We have handled the situations that input field of edited quiz name must not be empty and the start date must be before the due date.



Hình 5: Edit Quiz Form

When the user selects the delete option, the delete quiz form will pop up in the center of the screen.



Hình 6: Delete Quiz Pop-up



4.1.3 Question Page

The screenshot shows a web-based question creation interface. At the top, there's a navigation bar with 'Your courses > Web Programming > Practice Final Examination'. A red arrow labeled '1' points to the number '1' in a yellow box next to the 'Add a question' button. Below this is a form for creating a new question. A red arrow labeled '2' points to the number '2' in a yellow box next to the question description input field. The form includes fields for 'Question:', 'Option A*', 'Option B', 'Option C', 'Option D', and buttons for 'Cancel' and 'Submit'. A red arrow labeled '3' points to the number '3' in a yellow box next to the 'Submit' button. Below the main form, there are two questions listed. A red arrow labeled '4' points to the number '4' in a yellow box next to the first question. The first question is about the use of the `isset()` function in PHP. A red arrow labeled '5' points to the number '5' in a yellow box next to the 'View results' button. A red arrow labeled '6' points to the number '6' in a yellow box next to the 'Download PDF' button. The second question is about the output of a PHP program. A red arrow labeled '7' points to the number '7' in a yellow box next to the edit icon for this question.

Hình 7: Question Page

No	Label	Annotation
1	Add a question	Clicking on the button, the system will show the add question form.
2	Add question form	User enter the all the information of new question included (1) Question's description. (2) Four options details. (3) Level of question (Easy, Medium, Hard). Note that: In this form, we specify that Question A is the correct answer in default.
3	Submit	Clicking on the button, confirm creating new question.
4	Cancel	Clicking on the button, cancel creating new question.
5	View results	Clicking on the button, the system will navigate user to the Result page, which contains all marks of students did the quiz.
6	Download PDF	Clicking on the button, the system will navigate the user to print PDF page.
7	Edit Icon	Clicking on the button, the system will show the edit question form. The user can change all information of questions (descriptions, options and level).



When the user selects the editing option, the editing question form will pop up in the center of the screen. We have handled the situations that all input fields of edited question must not be empty, if the user wants to update it.

Question 1: What is the use of iset() Function in PHP?
A. The iset() function is used to check whether variable is set or not

Question:

Which PHP statement will give output as \$x on the screen?

Medium

Option A*
echo "\$x";

Option B
echo \$\$x;

Option C
echo "/\$x";

Option D
echo "\$x";

Delete Cancel Submit

Hình 8: Edit Question Form

At this page, the user can check again all their questions created in the quiz and course. Then they can download the file in PDF.

localhost:8080/teacher/print/

Web Programming
Practice Final Examination

Download PDF

Question 1: What is the use of iset() Function in PHP?
A. The iset() function is used to check whether variable is set or not
B. The iset() function is used to check whether the variable is free or not
C. The iset() function is used to check whether the variable is string or not
D. None of the above

Question 2: What will be the output of the following program? <?php \$var1 = "Hello"; \$var2 = "World"; echo \$var1\$var2; ?> ?
A. HelloWorld
B. \$var1\$var2
C. Hello World
D. None of the above

Question 3: Which PHP statement will give output as \$x on the screen??
A. echo "\$x";
B. echo \$\$x;
C. echo '\$x';
D. echo \$x;

Hình 9: Download PDF Feature



4.1.4 Result Page

The screenshot shows a web browser window with the URL `localhost:2080/teacher/?page=result`. The page title is "ADMIN PAGE". The main content area has a green header bar with the text "Your courses > Web Programming > Practice Final Examination > View result". Below this is a table with student marks:

No.	Name	Score
1	Nam Võ	50
2	Sáng Nguyễn	40
3	Đạt Võ	30
4	Linh Nguyễn	50
5	Tân Trần	30

At the bottom of the table, there are navigation buttons: '<', '1' (highlighted in red), and '>'. A copyright notice "Copyright 2021" is visible at the bottom of the page.

Hình 10: Result Page

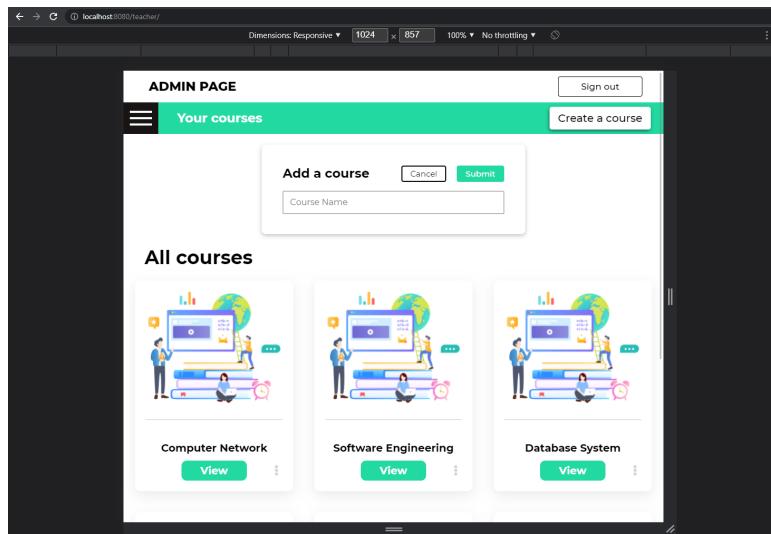
At this page, the user (teacher) can view all marks of the students who have done their specific quiz.

The score is calculated based on the level of the question. The higher level question is, the higher score is.

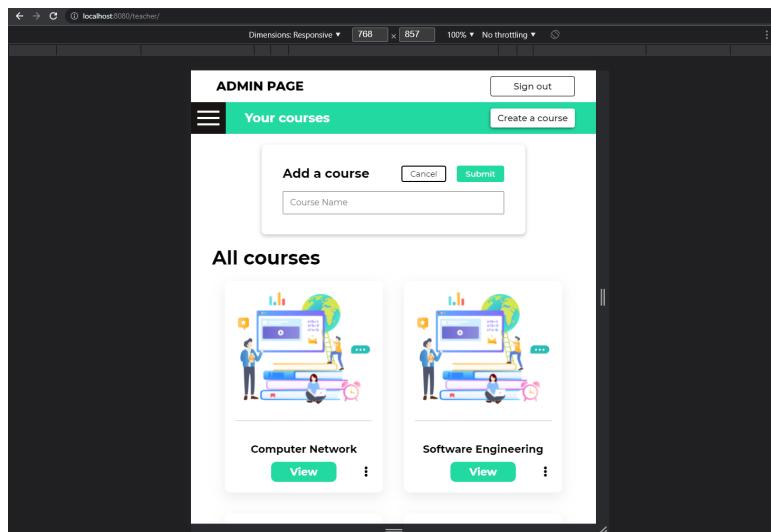


4.1.5 Responsive Pages

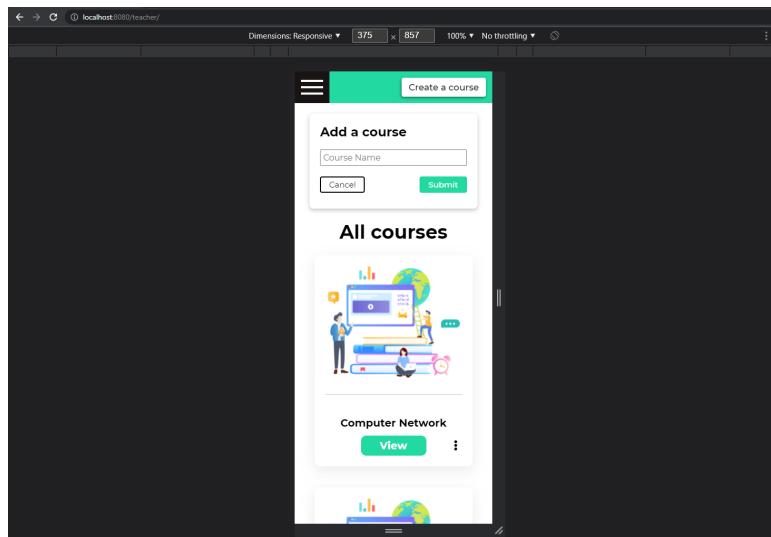
4.1.5.a Course Page



Hình 11: Course Page in Laptop view

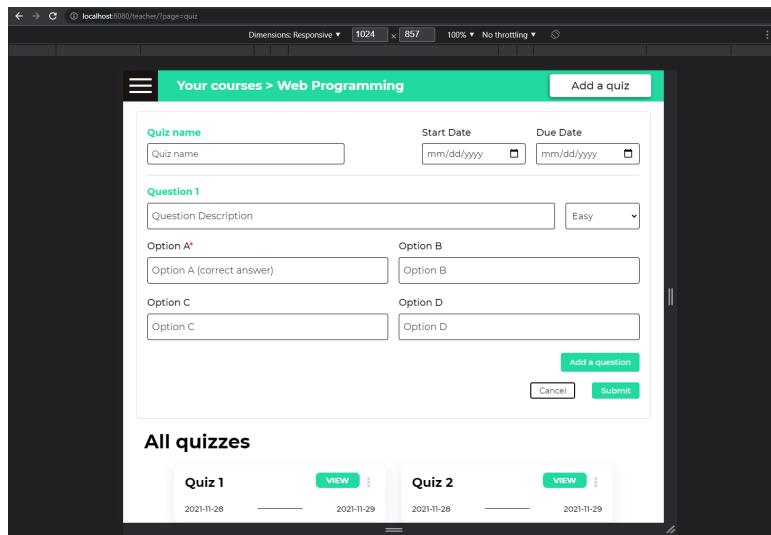


Hình 12: Course Page in Tablet view



Hình 13: Course Page in Mobile view

4.1.5.b Quiz Page



Hình 14: Quiz Page in Laptop view



The screenshot shows a tablet view of a quiz management interface. At the top, there's a header bar with the text "Your courses > Web Programming" and a "Add a quiz" button. Below this is a form for creating a new quiz, with fields for "Quiz name", "Start Date", and "Due Date". Underneath the form is a section titled "Question 1" containing a question description, a difficulty level dropdown set to "Easy", and four options labeled A, B, C, and D. Option A is highlighted as the correct answer. At the bottom of the form are "Add a question", "Cancel", and "Submit" buttons. Below the form, there's a section titled "All quizzes" listing four quizzes: Quiz 1, Quiz 2, Quiz 3, and Practice Final Examination, each with a "VIEW" button and a more options menu.

Hình 15: Quiz Page in Tablet view

The screenshot shows a mobile view of the quiz management interface. It features a similar header and quiz creation form as the tablet view. Below the form, the "All quizzes" section is displayed with the same four quizzes: Quiz 1, Quiz 2, Quiz 3, and Practice Final Examination, each with a "VIEW" button and a more options menu. At the bottom of the screen, there's a copyright notice: "Copyright 2021".

Hình 16: Quiz Page in Mobile view



The screenshot shows a mobile-optimized quiz creation interface. At the top right is a green button labeled "Add a quiz". Below it is a "Quiz name" field with a placeholder "Quiz name". Underneath are fields for "Start Date" and "Due Date", both in "mm/dd/yyyy" format. The next section is titled "Question 1" and contains a "Question Description" field and a dropdown menu set to "Easy". Below this are four option fields: "Option A*" containing "Option A (correct answer)", "Option B", "Option C", and "Option D".

Hình 17: Add Quiz Form in Mobile view

4.1.5.c Question Page

The screenshot shows a laptop view of a question page. At the top, there's a navigation bar with "ADMIN PAGE", "Your courses > Web Programming > Practice Final Examination", and a "Sign out" button. Below that is a "Add a question" button. The main area has a "Question" form with a "Question Description" field, a difficulty dropdown set to "Easy", and four option fields: "Option A*" (correct answer), "Option B", "Option C", and "Option D". At the bottom of the form are "Cancel" and "Submit" buttons, along with "View results" and "Download PDF" links. A modal window at the bottom displays a question about the use of the isset() function in PHP, with four options: A (correct), B, C, and D.

Hình 18: Question Page in Laptop view



The screenshot shows a tablet view of a web application. At the top, there's a navigation bar with 'ADMIN PAGE', 'Your courses > Web Programming > Practice Final Examination', and a 'Sign out' button. Below this is a form for creating a new question. The form includes fields for 'Question Description' (set to 'Easy'), 'Option A' (labeled 'correct answer'), 'Option B', 'Option C', and 'Option D'. Buttons for 'Cancel' and 'Submit' are at the bottom right. Below the form, there are three questions listed:

- Question 1:** What is the use of `isSet()` function in PHP?
 - A: The `isSet()` function is used to check whether variable is set or not
 - B: The `isSet()` function is used to check whether the variable is free or not
 - C: The `isSet()` function is used to check whether the variable is string or not
 - D: None of the above
- Question 2:** What will be the output of the following program?

```
<?php $var1 = "Hello"; $var2 = "World"; echo "$var1$var2";?>
```

 - A: HelloWorld
 - B: "\$var1\$var2"
 - C: Hello World
 - D: None of the above
- Question 3:** Which PHP statement will give output as \$x on the screen?
 - A: echo "\\$x";

Hình 19: Question Page in Tablet view

The screenshot shows a mobile view of the same web application. The layout is more compact, with the three questions stacked vertically. The questions and options are identical to those in the tablet view.

Hình 20: Question Page in Mobile view



The screenshot shows a mobile browser interface with a dark theme. At the top, there's a navigation bar with icons for back, forward, and search. Below it, the URL is shown as "localhost:8080/Teacher/?page=question". The main content area has a title "Add a question" with a three-line menu icon to its left. The form itself has a light background. It contains a "Question:" field with "Question Description" placeholder text, a dropdown for "Easy" difficulty, and a section for "Option A*" which includes a checked checkbox for "Option A [correct answer]". Below this are sections for "Option B", "Option C", and "Option D", each with an input field. At the bottom of the form are "Cancel" and "Submit" buttons, along with "View results" and "Download PDF" links. A note at the bottom states "Question 1: What is the use of insert function in PHP?".

Hình 21: Add Question Form in Mobile view

4.1.5.d Result Page

The screenshot shows a laptop browser interface with a dark theme. The URL "localhost:8080/teacher/?page=result" is visible at the top. The main content area has a header "ADMIN PAGE" with a "Sign out" button. Below it, a breadcrumb navigation shows "Your courses > Web Programming > Practice Final Examination > View result". The central part of the page is a table with columns "Name" and "Score". The data in the table is as follows:

Name	Score
Nam Võ	50
Sáng Nguyễn	40
Đạt Võ	30
Linh Nguyễn	50
Tấn Trần	30

At the bottom of the page, there are three small icons (refresh, back, forward). The footer contains the text "Copyright 2021".

Hình 22: Result Page in Laptop view



Name	Score
Nam Võ	50
Sáng Nguyễn	40
Đạt Võ	30
Linh Nguyễn	50
Tân Trần	30

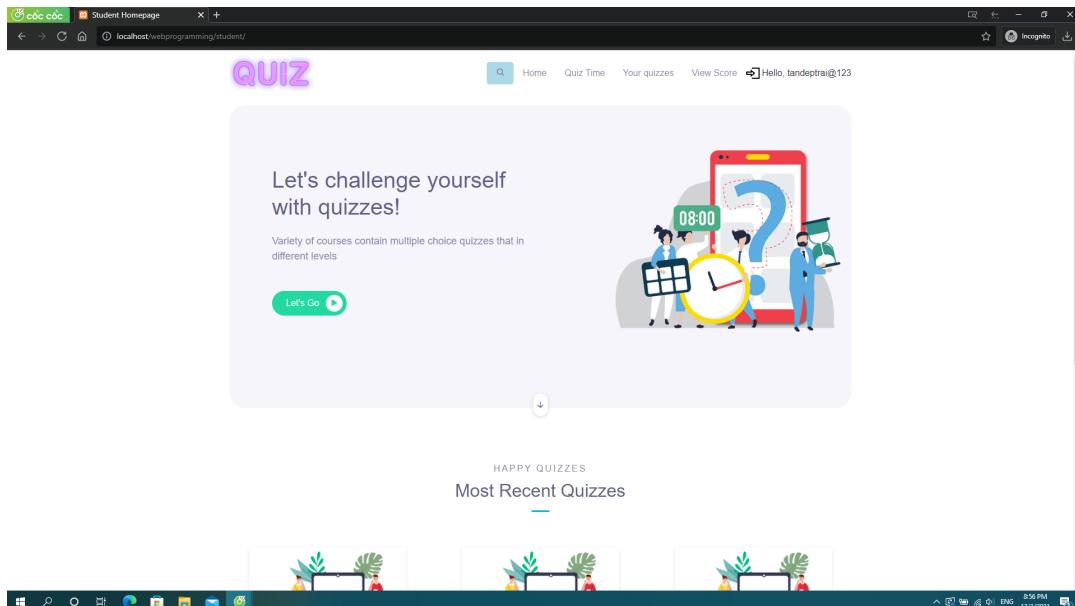
Hình 23: Result Page in Tablet view

Hình 24: Result Page in Mobile view

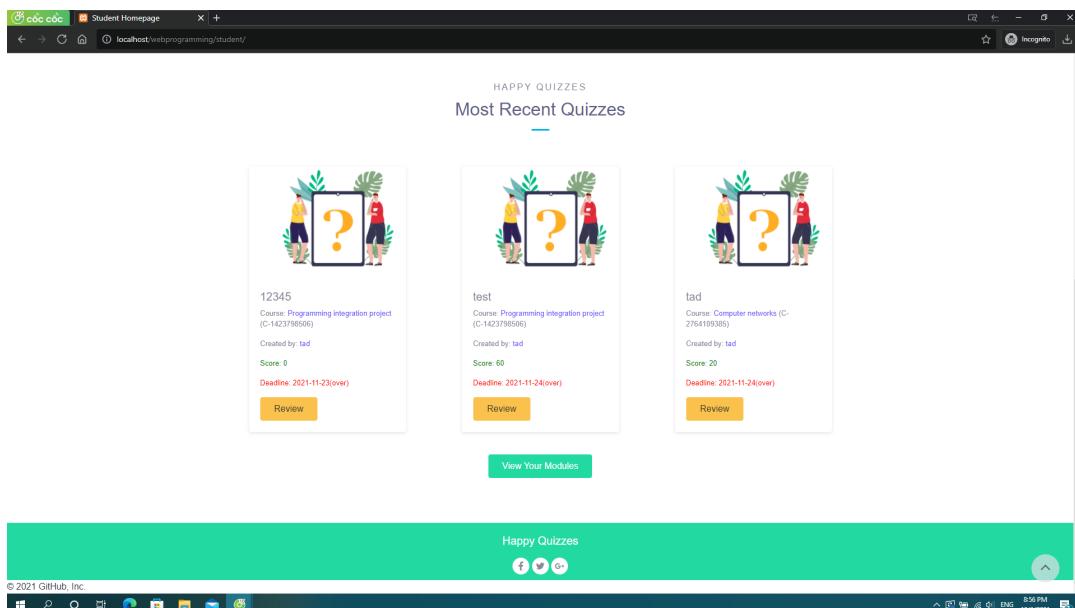


4.2 Student Implementation

4.2.1 Home Page



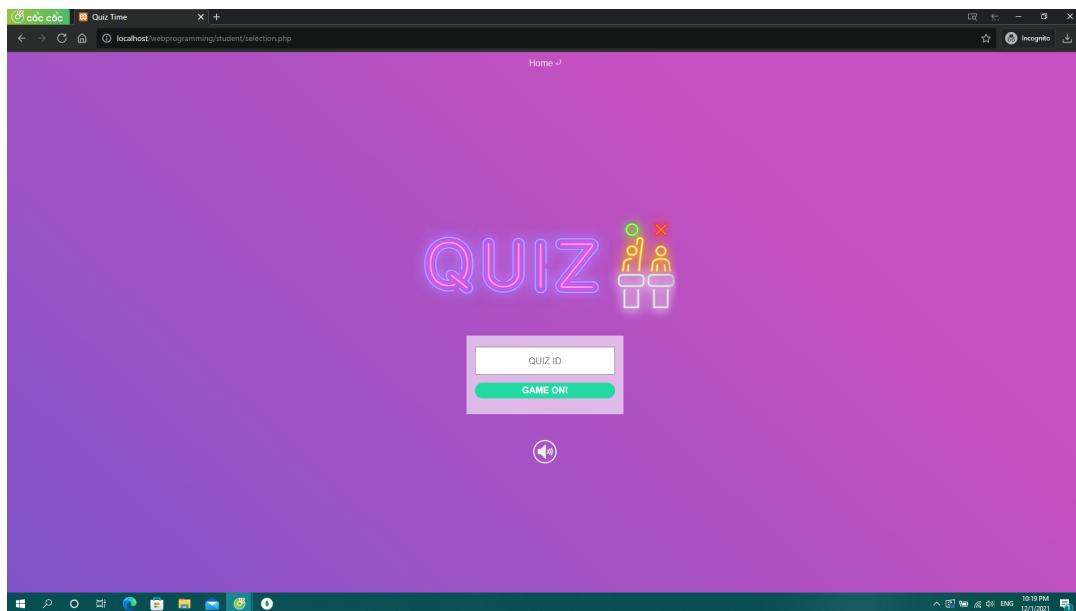
Hình 25: Course Page



Hình 26: Course Page



No	Label	Annotation
1	Sign out	Clicking on the button, Firstly, the system will navigate the user to login page. Secondly, the system will clear the user session.
2	Let's go	Clicking on the button, then enter the quiz ID and do the quiz.
3	Search	Click on the button and the system will
4	Your quizzes	Show all quizzes available.
5	View score	Clicking on the button, the system will navigate user to the view score page, which contains all quizzes done by that student.
6	Your recent quizzes	Show 3 most recent quizzes done by Student: (1) If the deadline of the quiz is over, the Review button is shown. (2) If the deadline is not over, the Do it button is shown.



Hình 27: Let's go



The screenshot shows a web browser window titled 'côc côc' with the URL 'localhost/webprogramming/student/module.php'. The page displays a student profile for 'Trần Nhật Tân' (STU-9101587797) and three quiz modules:

- Midterm exam**: Course: Data structures and algorithms (C-7960351842). Created by: Nguyen Minh Hoi. A green arrow button is next to it.
- Final exam**: Course: Data structures and algorithms (C-7960351842). Created by: Nguyen Minh Hoi. A green arrow button is next to it.
- Weekly exam**: Course: Computer networks (C-3750642018). A green arrow button is next to it.

Hình 28: All quizzes

4.2.2 Quiz Page

The screenshot shows a web browser window titled 'Quiz Time' with the URL 'localhost/webprogramming/student/game.php?id=Q-4238061597'. The page displays a quiz question:

QUIZ NAME Time Left: 06

1. For a binary search algorithm to work, it is necessary that the array (list) must be

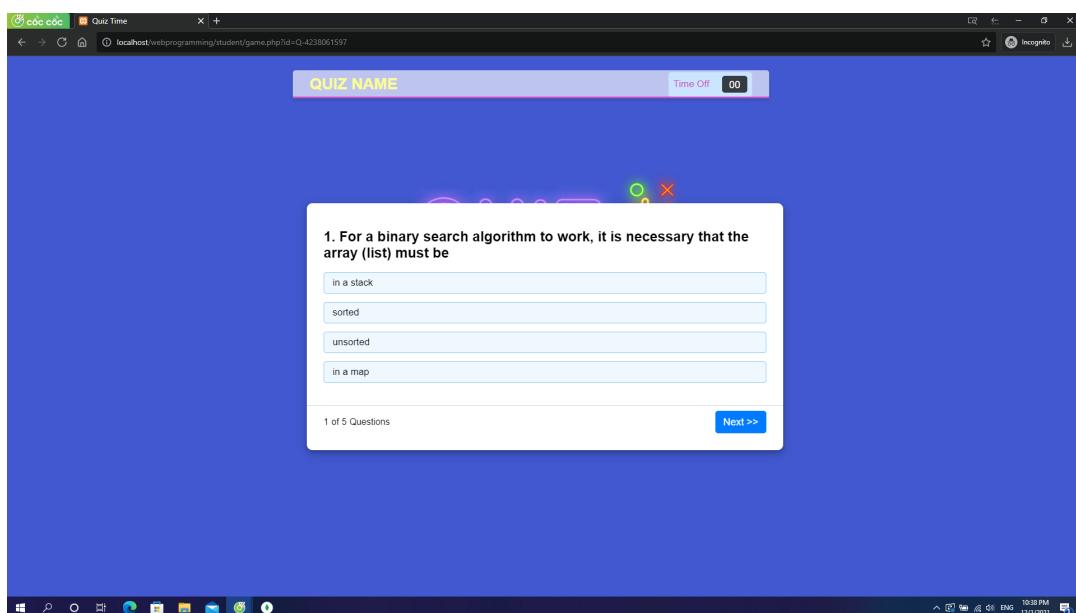
in a stack
 sorted
 unsorted
 in a map

1 of 5 Questions

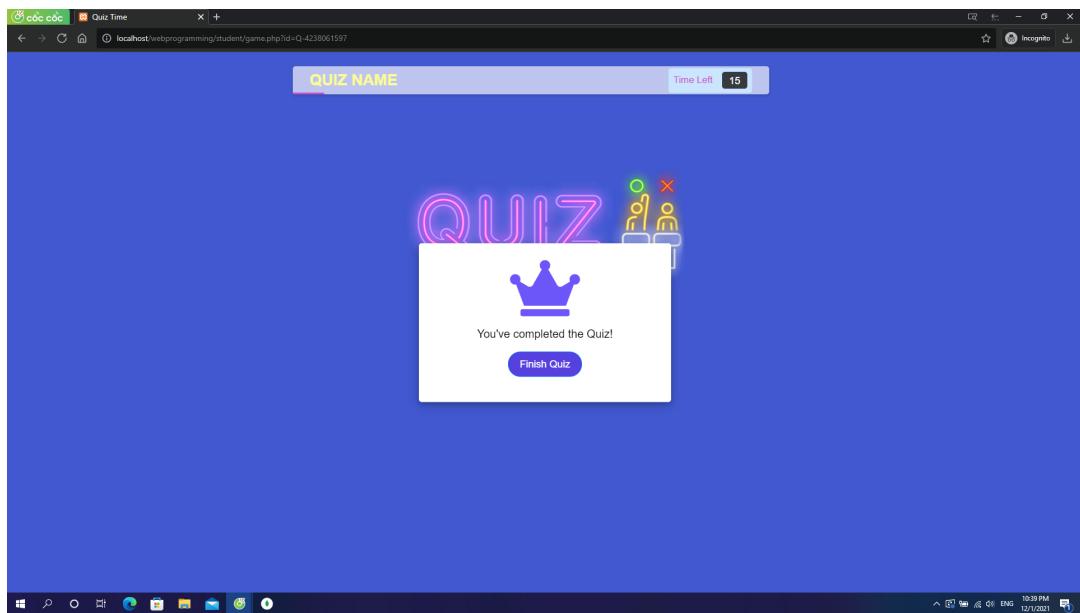
Hình 29: Quiz Page



No	Label	Annotation
1	Four answers	Clicking on one of the four answers, the Next button will appear, after 15 seconds, if no choice is chosen, the Next button appears and the answer is considered to be false.
2	Next	User choose a answer or no answer is chosen after 15 seconds (1) If the quiz does not end, the system will move to the next question. (2) Otherwise, the Finish button appears
3	Finish	Clicking on the button, the system will direct the user to the view score page



Hình 30: Next question



Hình 31: Finish quiz

4.2.3 View score Page

Hình 32: View score Page

If the deadline of the quiz is over, the review button appears, user is able to review the quiz. Otherwise, the Do it button appears, the user is able to do the quiz again.



Midterm exam
Data structures and algorithms
Nguyen Minh Hoi

Your total score is:
30/90

No.	Question	Correct answer	Level	Score
1	For a binary search algorithm to work, it is necessary that the array (list) must be	sorted	Easy	10
2	Maximum number of nodes in a binary tree with height k, where root is height 0, is	$2^k(k+1) - 1$	Easy	10
3	Which of the following is not possible with an array in C programming language	Dynamic allocation	Hard	30
4	What is not true about insertion sort?	None of the answers	Medium	20
5	Which of the below given sorting techniques has highest best-case runtime complexity	Selection sort	Medium	20

[Back to Home](#)

Hình 33: View quiz result

4.2.4 Responsive Pages

4.2.4.a Home Page

QUIZ

Let's challenge yourself with quizzes!

Recent quizzes

Most Recent Quizzes

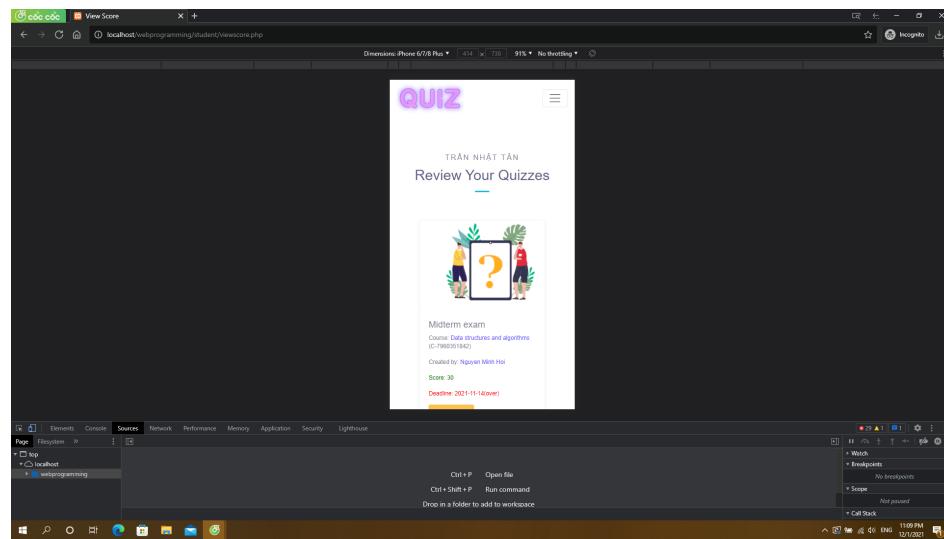
- Midterm exam
- 12345
- Test

Dimensions: iPad Pro | 1024 x 1366 | 49% | No throttling | Incognito

File Elements Console Sources Network Performance Memory Application Security Lighthouse

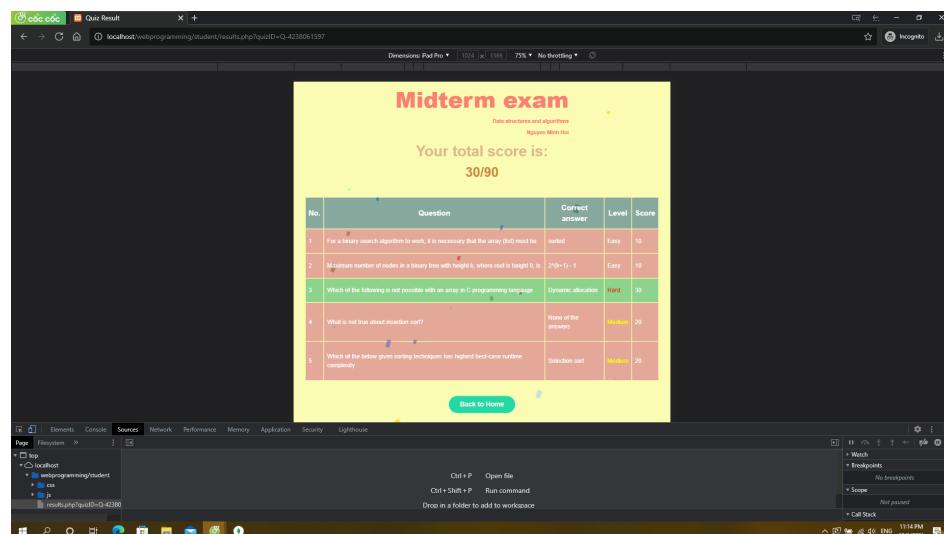
Ctrl + P Open file
Ctrl + Shift + P Run command
Drop in a folder to add to workspace

Hình 34: Course Page in Ipad view



Hình 35: Course Page in Iphone view

4.2.4.b Result Page



Hình 36: Result Page in Ipad view



5 Testing

5.1 What is Unit Testing and Integration Testing?

Unit testing is the process of testing program components, such as methods or object classes.

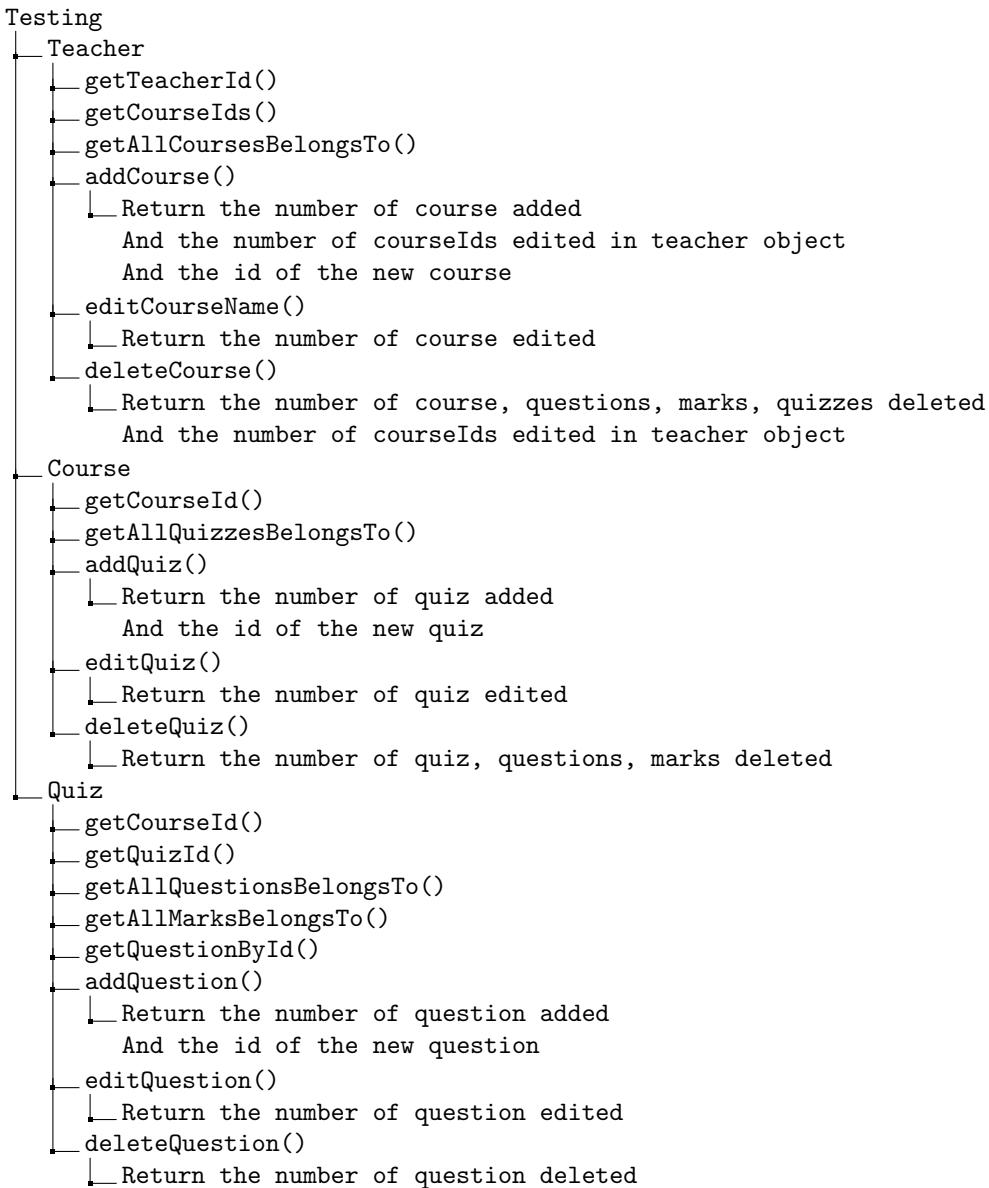
Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

In this testing section, we decided to use PHPUnit, which is a programmer-oriented testing framework for PHP.

5.2 Teacher Testing

5.2.1 Unit Test

First, we have to figure out what to test:





Second, based on this information, we have made our testing in three class named TeacherTest, CourseTest and QuizTest.

5.2.1.1 Teacher Test

(1) **setUp()** used to create the same Teacher object with teacherId = ‘TC-7320685914’ for all tests.

```
1  public function setUp(): void
2  {
3      // Create the same teacher object for all tests:
4      $this->teacher = new Teacher('TC-7320685914');
5 }
```

(2) **testGetCourseIds()** checks if the id of courses created by teacher is corrected.

```
1  public function testGetCourseIds()
2  {
3      $this->assertEquals($this->teacher->getCourseIds(), array("C-2490578136", "C-0527368149"));
4 }
```

(3) **testGetCourses()** checks returned courses whether belongs to the teacher, whose id is ‘TC-7320685914’ or not.

```
1  public function testGetCourses()
2  {
3      $courses = $this->teacher->getAllCoursesBelongsTo();
4      $coursesIds = array();
5      foreach ($courses as $course) {
6          array_push($coursesIds, $course->courseId);
7      }
8      $this->assertEquals($coursesIds, array("C-2490578136", "C-0527368149"));
9 }
```

(4) **testAddCourse()** checks adding course method:

```
1  public function testAddCourse()
2  {
3      $insertCourseResult = $this->teacher->addCourse('Web Programming');
4      $this->assertEquals($insertCourseResult[0], [1, 1]);
5
6      return $insertCourseResult[1];
7 }
```

To pass the test, the returned result must include 1 (number of added course) and 1 (number of updating teacher’s courseIds).

(5) **testEditCourse()** checks editing course method:

```
1 /**
2 * @depends testAddCourse
3 */
4 public function testEditCourse($courseId)
5 {
6     $editCourseName = $this->teacher->editCourseName($courseId, 'Data Structure');
7     $this->assertEquals($editCourseName, 1);
8
9     return $courseId;
10 }
```



To pass the test, the returned result must include 1 (number of update operation of course).

(6) **testDeleteCourse()** checks deleting course method:

```
1  /**
2   * @depends testEditCourse
3   */
4  public function testDeleteCourse($courseId)
5  {
6      $deleteCourseResult = $this->teacher->deleteCourse($courseId);
7      $this->assertEquals($deleteCourseResult, [0, 0, 0, 1, 1]);
8 }
```

To pass the test, the returned result must contain 0 (number of deleted mark), 0 (number of deleted question), 0 (number of deleted quiz), 1 (number of deleted course) and 1 (number of update operation of teacher'courseIds)

5.2.1.2 CourseTest

(1) **setUp()** helps us to create the same Course object, where teacherId = ‘TC-7320685914’ and courseId = ‘C-2490578136’ for all tests.

```
1  public function setUp(): void
2  {
3      // Create the same course object for all tests:
4      $this->course = new Course('C-2490578136', 'TC-7320685914');
5 }
```

(2) **testGetQuizzes()** used to check the returned quizzes whether belongs to the course, whose courseId = ‘C-2490578136’ or not.

```
1  public function testGetQuizzes()
2  {
3      $quizzes = $this->course->getAllQuizzesBelongsTo();
4      $quizzesIds = array();
5      foreach ($quizzes as $quiz) {
6          array_push($quizzesIds, $quiz->quizId);
7      }
8      $this->assertEquals($quizzesIds, array('Q-4670135892', 'Q-2790518436'));
9 }
```

(3) **testAddQuiz()** checks adding quiz method:

```
1  public function testAddQuiz()
2  {
3      $addQuizResult = $this->course->addQuiz(
4          "Midterm\u20d7Chemistry",
5          "2021-11-30",
6          "2021-12-1",
7      );
8      $this->assertEquals($addQuizResult[0], 1);
9      return $addQuizResult[1];
10 }
```

To pass the test, the returned result must include 1 (number of added quiz).

(4) **testEditQuiz()** checks editing quiz method:

```
1  /**
2   * @depends testAddQuiz
```



```
3     */
4     public function testEditQuiz($quizId)
5     {
6         $editQuizResult = $this->course->editQuiz(
7             $quizId,
8             "Final_Web-Programming",
9             "2021-11-30",
10            "2021-12-1",
11        );
12        $this->assertEquals($editQuizResult, 1);
13        return $quizId;
14    }
```

To pass the test, the returned result must include 1 (number of edited quiz).

(5) **testDeleteQuiz()** checks deleting quiz method:

```
1     /**
2      * @depends testEditQuiz
3      */
4     public function testDeleteQuiz($quizId)
5     {
6         $deleteQuizResult = $this->course->deleteQuiz($quizId);
7         $this->assertEquals($deleteQuizResult, [0, 0, 1]);
8     }
```

To pass the test, the returned result must include 0 (number of deleted mark), 0 (number of deleted question) and 1 (number of deleted quiz).

5.2.1.3 QuizTest

(1) **setUp()** helps us to create the same Quiz object, where courseId = 'C-2490578136' and quizId = 'Q-4670135892' for all tests.

```
1     public function setUp(): void
2     {
3         // Create the same quiz object for all tests:
4         $this->quiz = new Quiz('C-2490578136', 'Q-4670135892');
5     }
```

(2) **testGetQuestions()** checks getting all questions belongs to the quiz.

```
1     public function testGetQuestions()
2     {
3         $questions = $this->quiz->getAllQuestionsBelongsTo();
4         $checkQuestions = true;
5         $numberOfQuestions = 0;
6         foreach ($questions as $question) {
7             if ($question->quizId != $this->quiz->getQuizId() || $question->courseId
8                 != $this->quiz->getCourseId()) {
9                 $checkQuestions = false;
10            }
11            $numberOfQuestions++;
12        }
13        $this->assertEquals($checkQuestions, true);
14        $this->assertEquals($numberOfQuestions, 5);
15    }
```

To pass the test, it needs to satisfy two conditions such as "The number of returned questions is 5" and "All questions belongs to the quiz".



(3) **testGetMarks()** checks getting all marks belongs to the quiz.

```
1  public function testGetMarks()
2  {
3      $marks = $this->quiz->getAllMarksBelongsTo();
4      $checkMarks = true;
5      $numberOfMarks = 0;
6      foreach ($marks as $mark) {
7          if ($mark->quizId != $this->quiz->getQuizId()) {
8              $checkMarks = false;
9          }
10         $numberOfMarks++;
11     }
12     $this->assertEquals($checkMarks, true);
13     $this->assertEquals($numberOfMarks, 10);
14 }
```

To pass the test, it needs to satisfy two conditions such as "The number of returned marks is 10" and "All marks belongs to the quiz".

(4) **testAddQuestion()** checks adding question method:

```
1  public function testAddQuestion()
2  {
3      $insertQuestionResult = $this->quiz->addQuestion(
4          "What is the correct format for aligning written content to the center of
           the page in CSS?", 
5          "Text-align:center;",
6          "Font-align:center;",
7          "Text:align-center;",
8          "Font:align-center;",
9          2,
10     );
11     $this->assertEquals($insertQuestionResult[0], 1);
12
13     return $this->quiz->getQuestionById($insertQuestionResult[1]);
14 }
```

To pass the test, the inserted results should be 1 (number of added question).

(5) **testEditQuestion()** checks editing question method:

```
1  /**
2  * @depends testAddQuestion
3  */
4  public function testEditQuestion($question)
5  {
6      $updatedQuestionResult = $this->quiz->editQuestion(
7          $question->_id,
8          "What is the correct format for a div?",
9          "Div-id=example",
10         "Div_id='example'",
11         "Div='example'",
12         "Div.example",
13         1,
14     );
15     $this->assertEquals($updatedQuestionResult, 1);
16     return $question;
17 }
```

To pass the test, the edited results should be 1 (number of edited question).

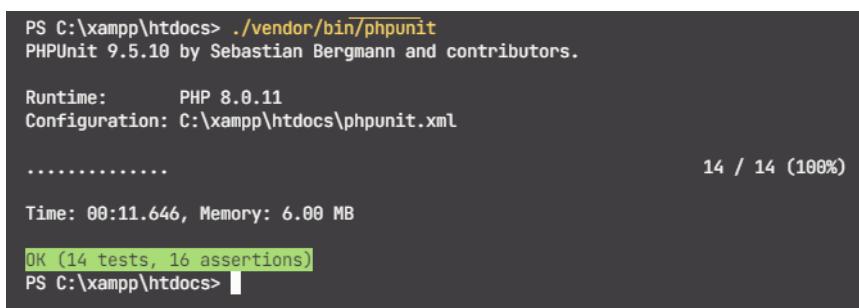
(6) `testDeleteQuestion()` to check deleting question:

```
1  /**
2   * @depends testEditQuestion
3   */
4  public function testDeleteQuestion($question)
5  {
6      $deleteQuestionResult = $this->quiz->deleteQuestion($question->_id);
7      $this->assertEquals($deleteQuestionResult, 1);
8  }
```

To pass the test, the deleted results should be 1 (number of deleted question).

5.2.1.4 Result of Unit Test

After created all tests, we run the test by entering `"./vendor/bin/phpunit"` in terminal.



```
PS C:\xampp\htdocs> ./vendor/bin/phpunit
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.0.11
Configuration: C:\xampp\htdocs\phpunit.xml

.....
14 / 14 (100%)

Time: 00:11.646, Memory: 6.00 MB

OK (14 tests, 16 assertions)
PS C:\xampp\htdocs>
```

Hình 37: Unit Test's result

According to the result, we have 14 tests and 16 assertions "PASSED" with Time: 00:11.646 and Memory: 6.00 MB.

Therefore, with unit test, our implementation worked as expected at this time.

5.2.2 Integration Test

In the integration test, we has combined three modules such as Teacher, Course and Quiz to expose faults in the interaction between them.

Firstly, `setUp()` used to create the same Teacher object, whose id = 'TC-7320685914'.

```
1  public function setUp(): void
2  {
3      // Create the same teacher object for all tests:
4      $this->teacher = new Teacher('TC-7320685914');
5  }
```

Secondly, `testAddOperation()` tests adding new course, quiz, question related to each other.

```
1  public function testAddOperation()
2  {
3      // 1. Add new course:
4      $insertCourseResult = $this->teacher->addCourse('Web Programming');
5      $this->assertEquals($insertCourseResult[0], [1, 1]);
6
7      // Create new course object based on new courseId:
8      $newCourseId = $insertCourseResult[1];
```



```
9     $newCourse = new Course($newCourseId, $this->teacher->getTeacherId());
10
11    // 2. Add new quiz to the new course:
12    $addQuizResult = $newCourse->addQuiz(
13        "Final_ Examination",
14        "2021-12-23",
15        "2021-12-24",
16    );
17    $this->assertEquals($addQuizResult[0], 1);
18
19    // Create new quiz object based on new quizId:
20    $newQuizId = $addQuizResult[1];
21    $newQuiz = new Quiz($newCourse->getCourseId(), $newQuizId);
22
23    // 3. Add new question to quiz:
24    $insertQuestionResult = $newQuiz->addQuestion(
25        "What does PDO stand for?",
26        "PHP Database Orientation",
27        "PHP Data Orientation",
28        "PHP Data Object",
29        "PHP Database Object",
30        1,
31    );
32    $this->assertEquals($insertQuestionResult[0], 1);
33
34    // Get id from new question:
35    $newQuestionId = $insertQuestionResult[1];
36
37    return [$newCourse, $newQuiz, $newQuestionId];
38 }
```

In this test,

- (1) We add a new course to the specific teacher.
- (2) We add a new quiz to the new course by course's id received by addition result on the first stage.
- (3) We add a new question to the quiz created on the second stage.

The returned result should be 1 (number of added course), 1 (number of added quiz) and 1 (number of added question) corresponding to the above flow to pass the test.

Thirdly, **testEditOperation()** tests editing course, quiz, question related to each other.

```
1 /**
2  * @depends testAddOperation
3 */
4 public function testEditOperation($addResult)
5 {
6     // Get value from testAddOperation():
7     $targetCourse = $addResult[0];
8     $targetQuiz = $addResult[1];
9     $targetQuestionId = $addResult[2];
10
11    // 1. Edit course:
12    $editCourseName = $this->teacher->editCourseName($targetCourse->getCourseId
13        (), 'Programming_Web');
14    $this->assertEquals($editCourseName, 1);
15
16    // 2. Edit quiz:
17    $editQuizResult = $targetCourse->editQuiz(
18        $targetQuiz->getQuizId(),
19        "Midterm_ Exam",
20        "2021-12-24",
```



```
20     "2021-12-25",
21 );
22 $this->assertEquals($editQuizResult, 1);
23
24 // 3. Edit question:
25 $updatedQuestionResult = $targetQuiz->editQuestion(
26     $targetQuestionId,
27     "What does PHP stands for?",
28     "Hypertext Preprocessor",
29     "Pretext Hypertext Preprocessor",
30     "Personal Home Processor",
31     "None of the above",
32     2,
33 );
34 $this->assertEquals($updatedQuestionResult, 1);
35
36 return [$targetCourse, $targetQuiz, $targetQuestionId];
37 }
```

In this test, we get course's id, quiz's id, question's id from `testAddOpeartion()` method. After that, based on this information, we make some edit on them.

The returned result must include 1 (number of edited course), 1 (number of edited quiz) and 1 (number of edited question).

Fourthly, `testDeleteOperation()` tests deleting course, quiz, question related to each other.

```
1 /**
2  * @depends testEditOperation
3 */
4 public function testDeleteOperation($editResult)
5 {
6
7     // Get value from testEditOperation():
8     $targetCourse = $editResult[0];
9     $targetQuiz = $editResult[1];
10    $targetQuestionId = $editResult[2];
11
12    // 1. Delete Testing:
13    // Delete course:
14    $deleteCourseResult = $this->teacher->deleteCourse($targetCourse->
15        getCourseId());
16    $this->assertEquals($deleteCourseResult, [0, 1, 1, 1, 1]);
17
18    return [$targetCourse, $targetQuiz, $targetQuestionId];
19 }
```

In this test, we use course's id received from `testEditOperation()` to delete the course.

Then we check if the result returns 0 deleted mark, 1 deleted question, 1 deleted quiz, 1 deleted course, 1 update of teacher's courseId.

Fifthly, `testGetOperation()` to check if `testDeleteOperation()` is successful.

```
1 /**
2  * @depends testDeleteOperation
3 */
4 public function testGetOperation($deleteResult)
5 {
6     // Get value from testDeleteOperation():
7     $targetCourse = $deleteResult[0];
8     $targetQuiz = $deleteResult[1];
9     $targetQuestionId = $deleteResult[2];
```



```
10 // Get all ids of courses created by teacher:  
11 $this->assertEquals($this->teacher->getCourseIds(), array("C-2490578136", "C-0527368149));  
12  
13 // Get all courses belongs to the teacher:  
14 $courses = $this->teacher->getAllCoursesBelongsTo();  
15 $coursesIds = array();  
16 foreach ($courses as $course) {  
17     array_push($coursesIds, $course->courseId);  
18 }  
19  
20 $this->assertEquals($coursesIds, array("C-2490578136", "C-0527368149"));  
21  
22 // Get all courses belongs to new course:  
23 $quizzes = $targetCourse->getAllQuizzesBelongsTo();  
24 $questions = $targetQuiz->getAllQuestionsBelongsTo();  
25 $this->assertEquals(count($quizzes->toArray()), 0);  
26 $this->assertEquals(count($questions->toArray()), 0);  
27  
28 return [$targetCourse, $targetQuiz, $targetQuestionId];  
29 }
```

To make sure that the previous testDeleteOpeartion() works as expected, we make a testing about get operation.

Lastly, **testFailedOperation()** to check some failed operations.

```
1 /**
2 * @depends testGetOperation
3 */
4 public function testFailedOperation($getOperation)
5 {
6     // Get value from testGetOperation():
7     $targetCourse = $getOperation[0];
8     $targetQuiz = $getOperation[1];
9     $targetQuestionId = $getOperation[2];
10
11    // 1. Edit the course doesn't exist in database:
12    $editCourseName = $this->teacher->editCourseName($targetCourse->getCourseId(),
13        'Web Programming');
14    $this->assertEquals($editCourseName, 0);
15
16    // 2. Edit the quiz doesn't exist in database:
17    $editQuizResult = $targetCourse->editQuiz(
18        $targetQuiz->getQuizId(),
19        "Midterm Exam",
20        "2021-12-24",
21        "2021-12-25",
22    );
23    $this->assertEquals($editQuizResult, 0);
24
25    // 3. Delete the quiz doesn't exist in database:
26    $deleteQuizResult = $targetCourse->deleteQuiz($targetQuiz->getQuizId());
27    $this->assertEquals($deleteQuizResult, [0, 0, 0]);
28 }
```

At this final step, we aim to test some failed situations because at this time the tested course, quiz and question were no longer in our database. Therefore, those operations will return 0 update results.



5.2.2.1 Result of Integration Test

After created all tests, we run the test by entering `./vendor/bin/phpunit` in terminal.

```
PS C:\xampp\htdocs> ./vendor/bin/phpunit
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.0.11
Configuration: C:\xampp\htdocs\phpunit.xml

.
.
.
19 / 19 (100%)

Time: 00:21.905, Memory: 6.00 MB

OK (19 tests, 30 assertions)
PS C:\xampp\htdocs> 
```

Hinh 38: Integration Test's result

We have 19 tests and 30 assertions "PASSED" with Time: 00:21.905 and Memory: 6.00 MB. In those tests, there are 5 tests, 14 assertions from Integration test and the others from Unit test. Therefore, our implementations worked as expected at this time.



5.3 Student Testing

5.3.1 Unit Test

(1) **testGetStudentId()** is used to make sure getting the right Student Id

```
1 public function testGetStudentId() : void {
2     $studentInfo = new StudentInfo();
3     $username = "stu.Gciv@gmail.com";
4
5     $expected = "STU-0195872364";
6     $actual = $studentInfo->getStudentId($username);
7     $message = "\n[Failure\u2014in\u2014test\u2014getStudentId]";
8
9     $this->assertEquals($expected, $actual, $message);
10 }
```

(2) **testGetTeacherUsername()** is used to make sure getting the right teacher's username

```
1 public function testGetTeacherUsername() : void {
2     $studentInfo = new StudentInfo();
3     $teacherId = "TC-4203517869";
4
5     $expected = "tc.wZAt@gmail.com";
6     $actual = $studentInfo->getTeacherUsername($teacherId);
7     $message = "\n[Failure\u2014in\u2014test\u2014getTeacherUsername]";
8
9     $this->assertEquals($expected, $actual, $message);
10 }
```

(3) **testGetCourseName()** is used to make sure getting the right course name

```
1 public function testGetCourseName() : void {
2     $studentInfo = new StudentInfo();
3     $courseId = "C-7960351842";
4
5     $expected = "Data\u2014structures\u2014and\u2014algorithms";
6     $actual = $studentInfo->getCourseName($courseId);
7     $message = "\n[Failure\u2014in\u2014test\u2014getCourseName]";
8
9     $this->assertEquals($expected, $actual, $message);
10 }
11 }
```

(4) **testGetQuizDueDate()** is used to make sure getting the right quiz due date

```
1 public function testGetQuizDueDate() : void {
2     $studentInfo = new StudentInfo();
3     $quizId = "Q-0329681754";
4
5     $expected = "2021-11-20";
6     $actual = $studentInfo->getQuizDueDate($quizId);
7     $message = "\n[Failure\u2014in\u2014test\u2014getQuizDueDate]";
8
9     $this->assertEquals($expected, $actual, $message);
10 }
11 }
```



5.3.2 Result of Unit Test

```
PS C:\xampp\htdocs\wp> ./vendor/bin/phpunit
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.0.12
Configuration: C:\xampp\htdocs\wp\phpunit.xml

....                                     4 / 4 (100%)

Time: 00:03.686, Memory: 6.00 MB

OK (4 tests, 4 assertions)
PS C:\xampp\htdocs\wp> █
```

Student unit test's result

According to the result, we have 4 tests and 4 assertions "PASSED" with Time: 00:03.686 and Memory: 6.00 MB. It means that the our implementation is working fine at this time.

6 Evaluation

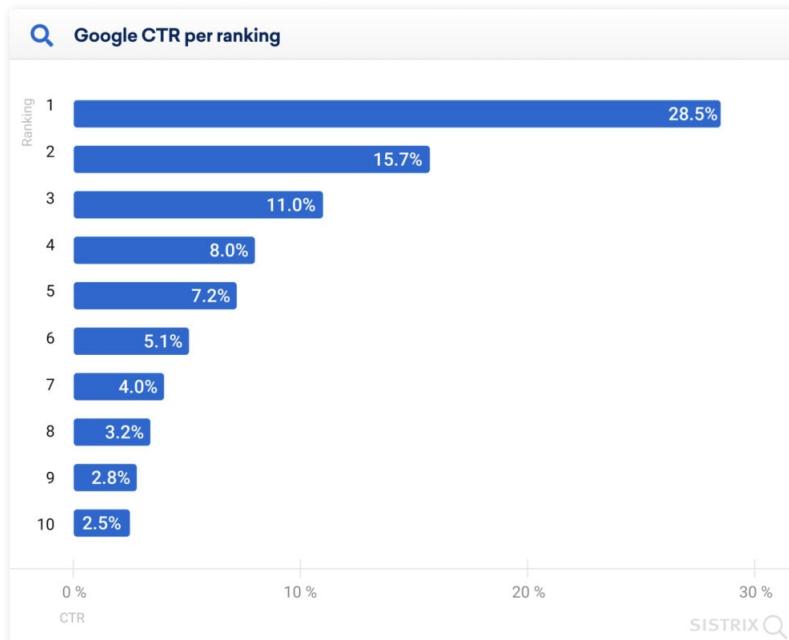
6.1 Search Engine Optimization (SEO)

6.1.1 What is SEO?

SEO is an abbreviation for "Search Engine Optimization." It refers to the process of improving your website so that it appears more prominently when people search for products or services related to your company on search engines. The higher the visibility of your pages in search results, the more likely you are to attract attention and attract new and existing customers to your company.

6.1.2 Why SEO is important

According to a 2020 study about Organic Click-Through-Rate (CTR), the number of web searchers who click on your result in the SERPs divided by the number of impressions your result receives, more than 70% of search engine users are more likely to click on one of the top 5 suggestions in the search engine results pages (SERPS).



2020 CTR by Ranking positions

To take advantage of this and gain more visitors, our QUIZ website needs to appear in one of the top positions. Furthermore, users will have more trust in our website if it shows in the top places.

In addition, efficient SEO practices will improve the user experience and usability of a website, have better social promotions, increases organic search engine traffic and the number of visitors, etc.

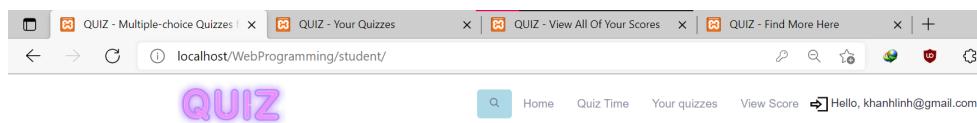


6.1.3 What we use to improve SEO.

1. Title tag

Title tags are a major factor in helping search engines understand what the page is about. A well-written title will be optimized for search engines so it can appeal to being scanned by robots and then searched for by humans. When searching for websites, online readers just scan a web page before they completely read everything. So, the better title will attract more users and increase CTR.

We named all the pages on our website with the same syntax: **QUIZ - page's purpose** which aids in the consistency of the web structure. This helps searchers understand the web system's overall function, as well as the function of each individual web page.



2. Keywords

Keywords are the words and phrases that people type into search engines, often known as "search queries" in SEO. Keywords are important because they are the linchpin between what people are searching for and the content we are providing to fill that need. Search engines use keywords to determine which content is relevant to a particular search query, and how the page should rank in searches for a particular term.

We chose simple short terms that users can type in and possible typos for our website keywords. When users search for a specific website, sometimes they enter its address into the search bar instead of the address bar. So, we also include that case in our keywords.

```
<meta name="keywords" content="quiz, quizz, quizzes, quiz.com, www.quiz.com, quiz.com website, online quiz, join quiz, do quiz quiz login, quiz student, quiz student login">
```

3. Description Tag

A meta description is a sentence or two that describes the website's content. It is important in SEO and widely used in search engine results pages (SERPs) to offer basic descriptions for a certain page. Search engines crawl that text to find relevant phrases that match a search query. Description of the website will offer more opportunities to add key information or "ad copy", increase CTR and gain more organic traffic. Pages with a meta description experience 5.8% more clicks than those that don't.

To capture users' attention, we use short sentences describing what we can offer them. These descriptions will also be a factor in their decision to choose our website over others.

```
<meta name="description" content="The effective online-quiz system for schools to create, manage and take online quizzes. Suitable for any devices · Sign up for free!">
```

4. Heading Tags (h1 - h6)

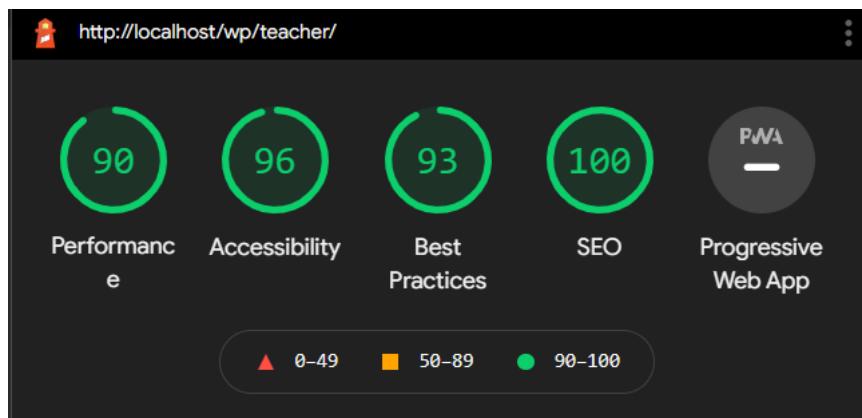
In search engine optimization, heading tags are critical for attracting visitors to your website. The headings are what people are shown after searching and determine whether or not

users click on that article. The h1 tags tell what the article is about. This is important, as it is the whole reason the user clicked on, and will stay on the article. The h2 and h3 tags organize the rest of the article into easy-to-read sections and also come up in searches.

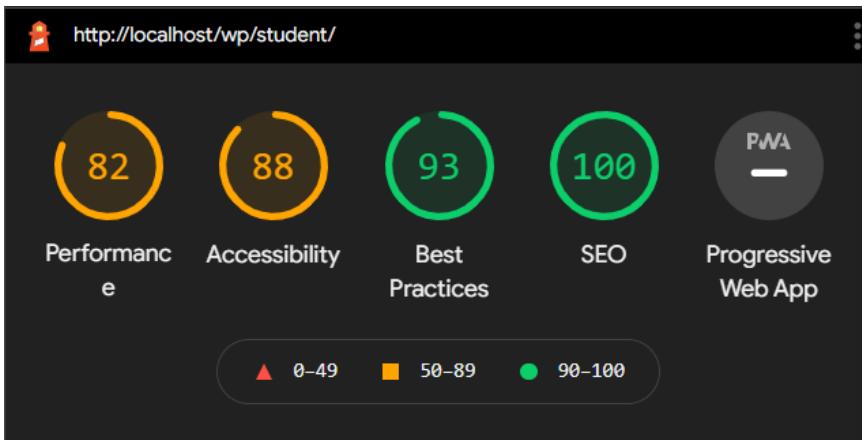
The heading tags on our web pages all contain words related to the topic of our website, and the majority of them contain our chosen keywords.

6.2 Website Rating Point

We have tested website rating point on 2 main pages (teacher homepage and student homepage) on Chrome's Lighthouse.



Teacher homepage rating point



Student homepage rating point



7 Conclusion

7.1 Source code and domain

This is our GitHub repository for this project: <https://github.com/tuniss1/WebProgramming>

7.2 Results review

Regarding the functionality of the website, the project has realized all the functions set out in the beginning, the functions work as expected.

Regarding the frontend, the website has a very reasonable design for users, suitable for the nature of a creating and doing quiz website, emphasizing the parts that users need to pay attention to. Users can easily navigate between pages, the functions of the website work smoothly, bringing comfort to users.

Regarding the backend, the website uses MongoDB's service. Data information is stored on MongoDB authentication and MongoDB to increase the security and stability of the database.

Overall, the project runs as expected. The completion rate is 100%.

7.3 What we learned?

At the end after we overcome our difficulty we have learned:

- How to teamwork.
- Time management skill.
- Make a professional meeting with group members.
- Making decisions about how a system should work.
- Logical thinking about the end-users experience.
- Cooperating using Github system.
- Fast-learning some new technologies.