GROUP PROJECT FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	WEBG301 - Project Web		
Submission date	1/6/2022	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Nguyen Duc Anh	Student ID	GCH17051
Class	GCH0805	Assessor name	Nguyen Dinh Tran Long

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

		Student's signature	ducanh

Grade

☐ **Summative Feedback:**   ☐ **Resubmission Feedback:**

Grade:	**Assessor Signature:**	**Date:**
Signature & Date:		

Contents

Project Web

Student Management System

I. Introduction

The article is a group report on the student management system, consisting of three parts. Part one introduces user requirements, ERD database design, and Site map. The second part shows the MVC pattern used in the project and screenshots of the website. The final part evaluates the site's strengths and weaknesses, and future improvements. In this project, the author only works with Student and Class function. Therefore, in this article, some part only introduces about things related to the above two functions.

II. User Requirement

As a/an	I want to…<perform some task>	So that I can…<achieve some goal>
Admin	Manage all account of all user	Create account for all Student, Teacher
	Manage all Course	Create New, Update information, delete, search
	Manage all Class	Create New, Update information, delete, search
	Manage all Student	Assign and remove student to class Update information, delete, search, create new
	Manage all Teacher	Create New, Update information, delete, search
	Manage all Room	Create New, Update information, delete, search
	Manage all Category	Create New, Update information, delete, search
Teacher	Check Course, Class, Room, Course Category, Student	See list course, class, student, Category and see detail course, class, Course Category Search Course, Class, Room, Course Category, Student, Teacher
Student	Check Course, Class, Room, Course Category,	See list course, class, Category and see detail course, class, Course Category

		Search Course, Class, Room, Course Category,

Figure 1: User requirement

III. System Design
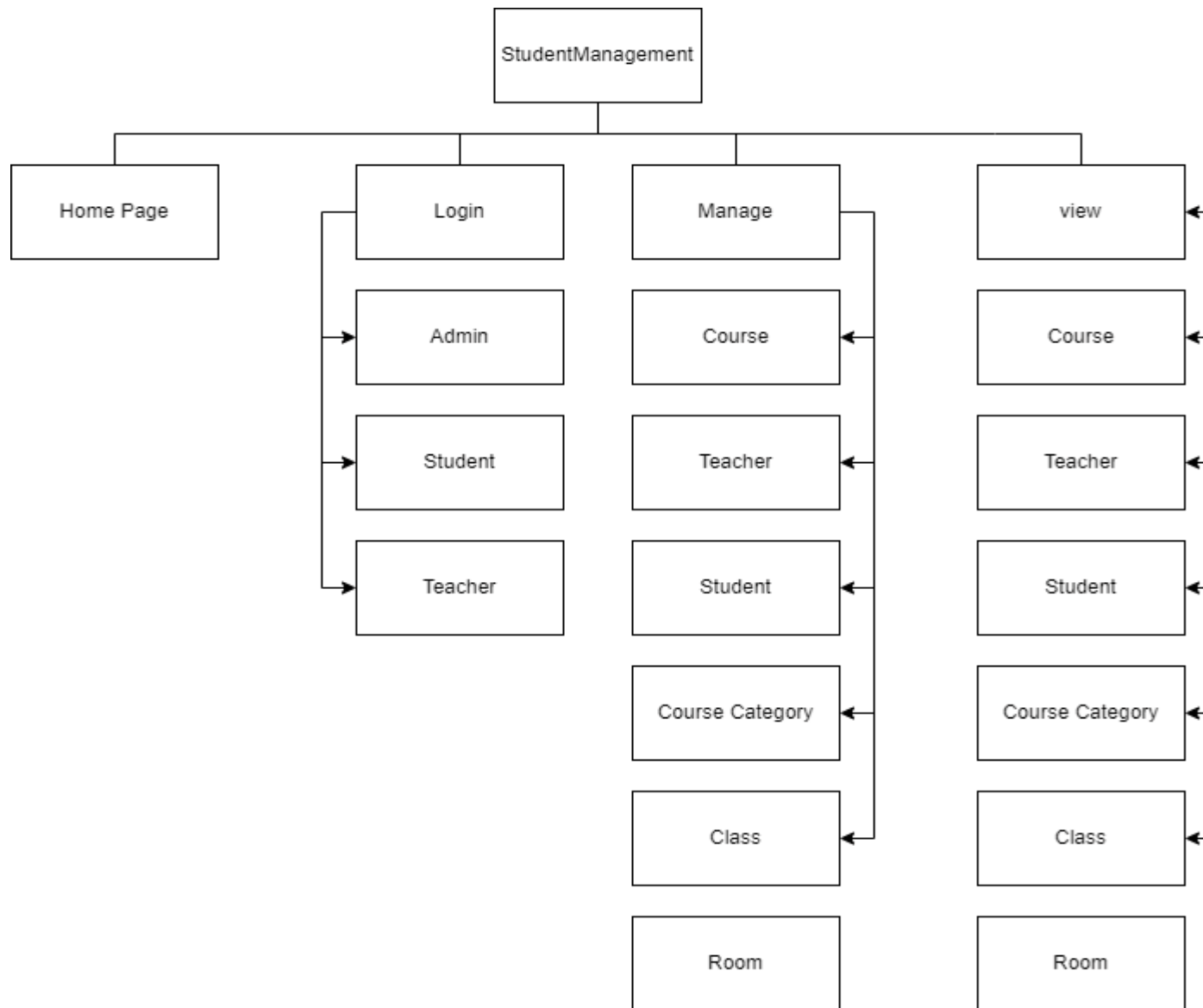### 1. Site map

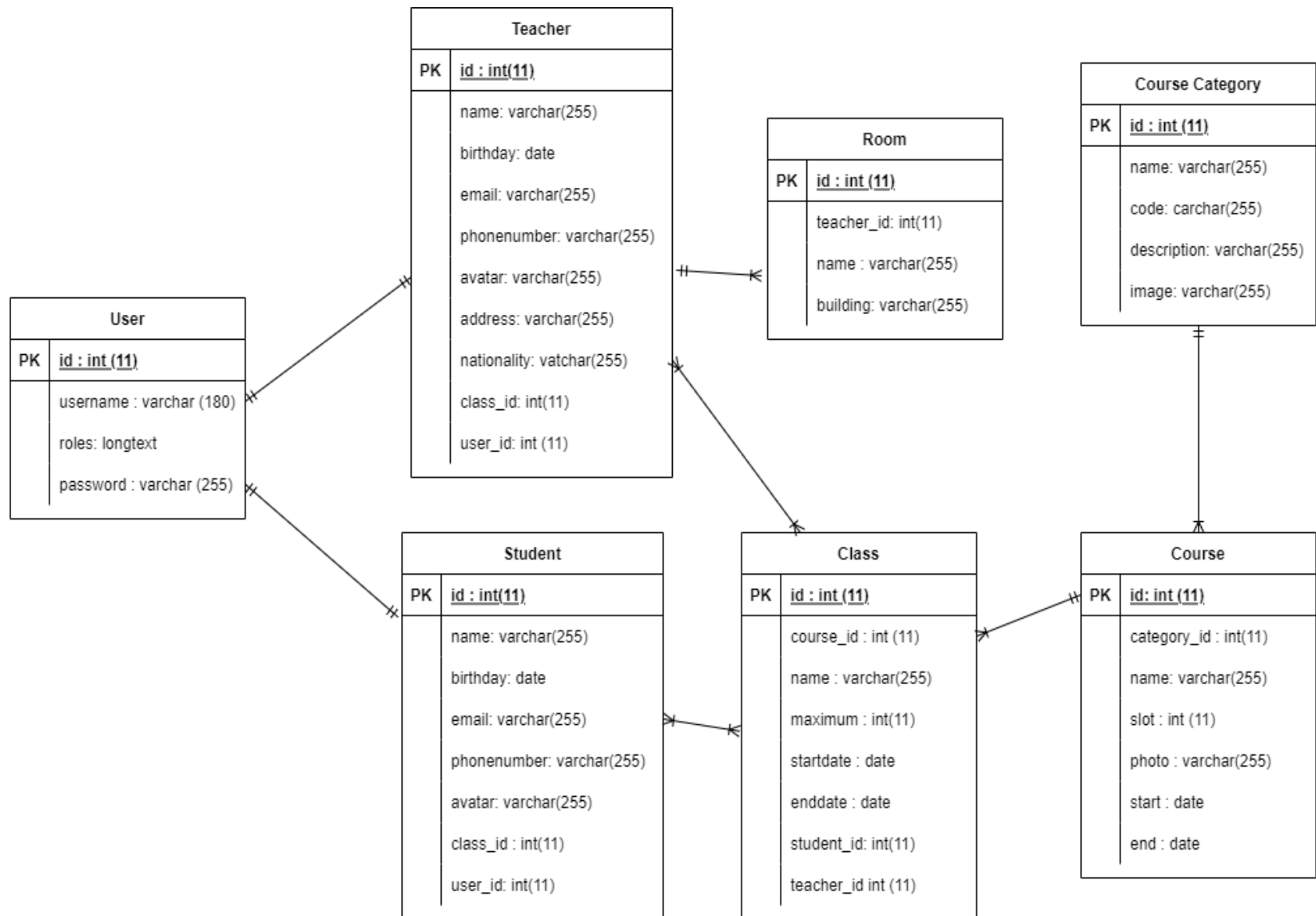Figure 2: Site map

2. Entity Relationship Diagram

Figure 3: ERD diagram

IV. Implementation
1. Sample Source Code

The system uses the Symfony framework, so the image below shows the directory of the whole project that has been created with the Symfony command. Obviously, Symfony has divided folders according to the MVC pattern: model and controller in the src folder, and view in the templates folder.
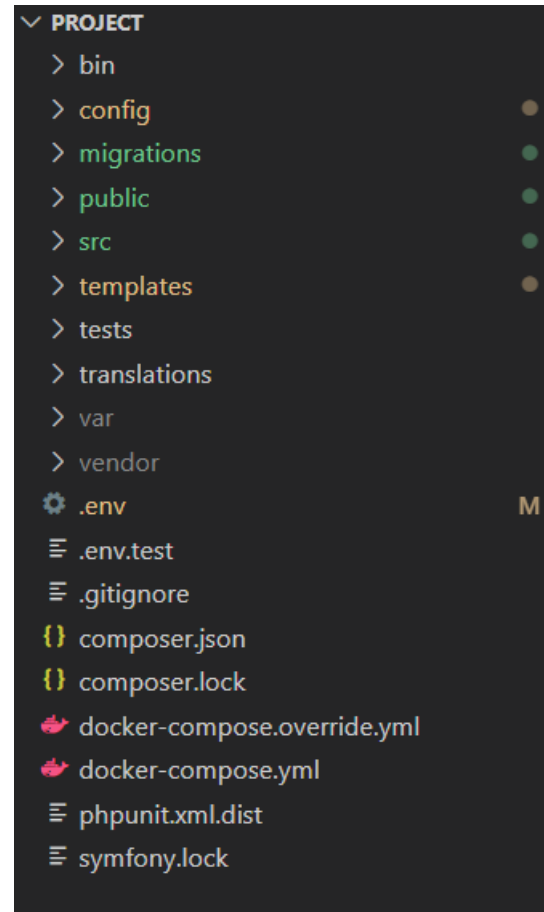
Figure 4: Source code folder
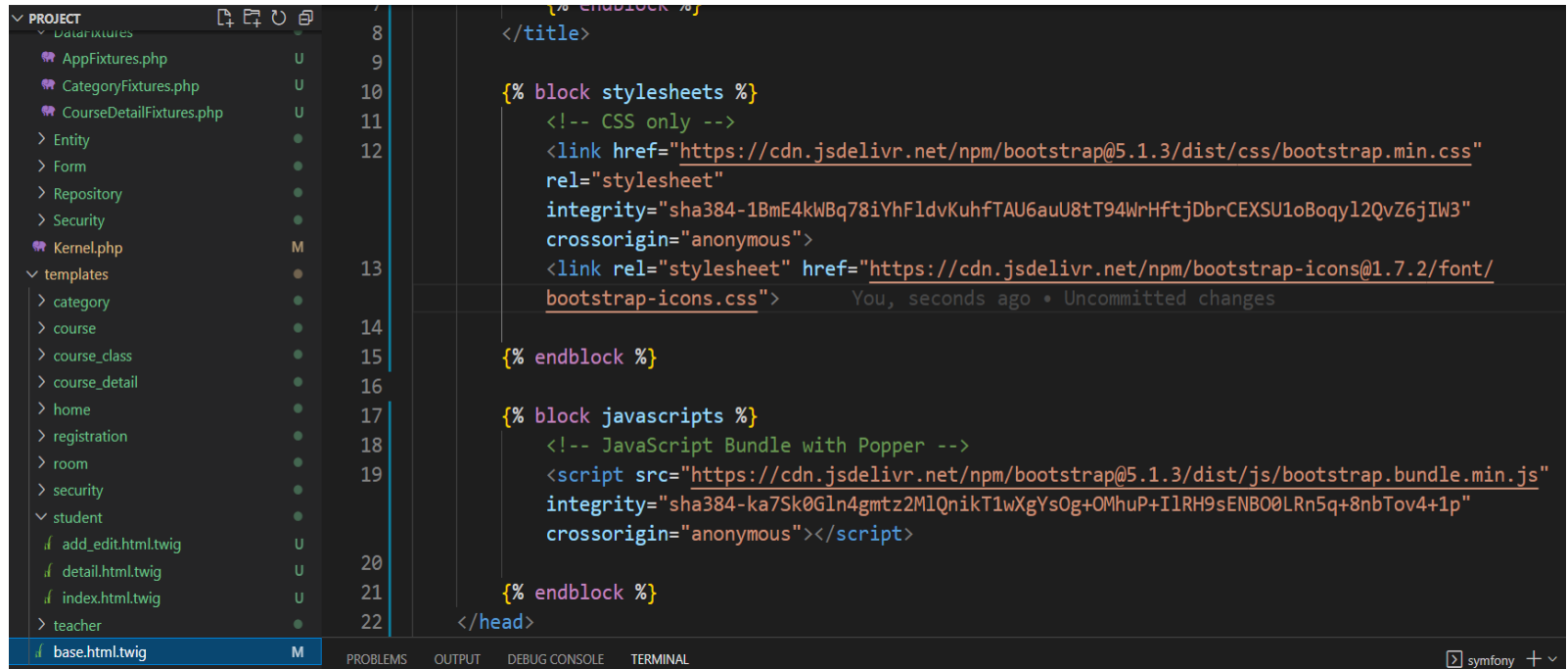
a. **View**

In the MVC model, the View represents the user interface. In symphony, this view is located in the "templates" folder.

Figure 5: View folder

The following are some example of view

Figure 6: base.html.twig

Figure 7: Student index

b. Controller

Basically, in the MVC model, the controller acts as a middle layer between the view and the model. It receives commands from the view and sends them to the model and vice versa.
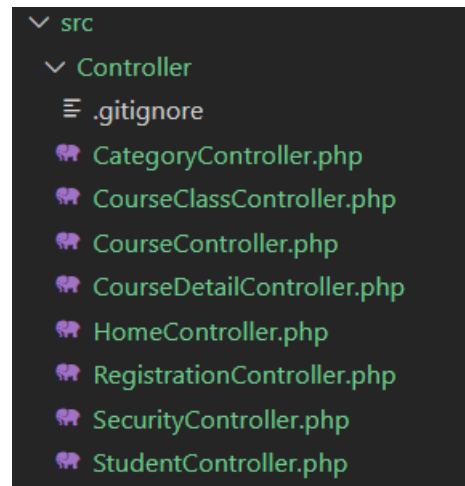
Figure 8: Controller folder

An example in the student's controller, two functions studentIndex and studentDetail are responsible for getting data from the Student table in the database and rendering it in the view.
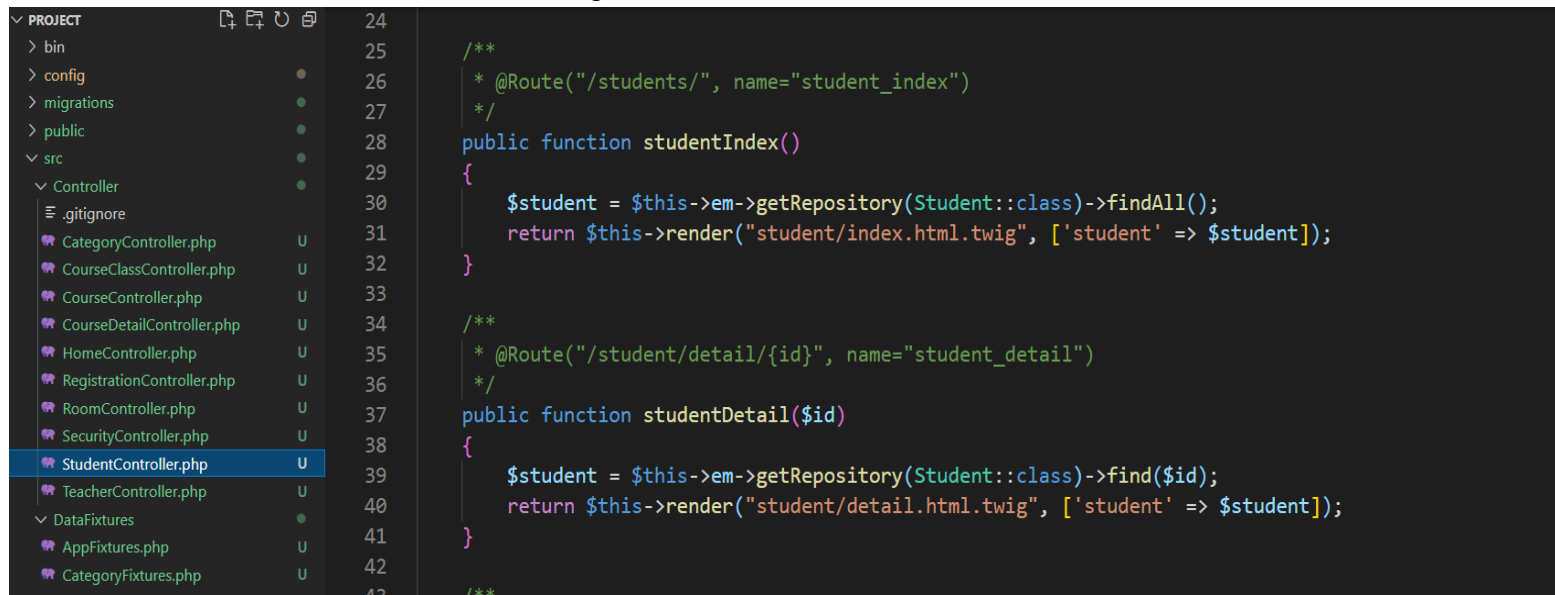


```php
/**
 * @Route("/students/", name="student_index")
 */
public function studentIndex()
{
    $student = $this->em->getRepository(Student::class)->findAll();
    return $this->render("student/index.html.twig", ['student' => $student]);
}

/**
 * @Route("/student/detail/{id}", name="student_detail")
 */
public function studentDetail($id)
{
    $student = $this->em->getRepository(Student::class)->find($id);
    return $this->render("student/detail.html.twig", ['student' => $student]);
}
```

Figure 9: Student controller

c. Model

In simple terms, in the MVC pattern, this model handles and accesses operations in the database. In Symfony, the Model can be auto-generated with commands such as 'symfony make: entity'.
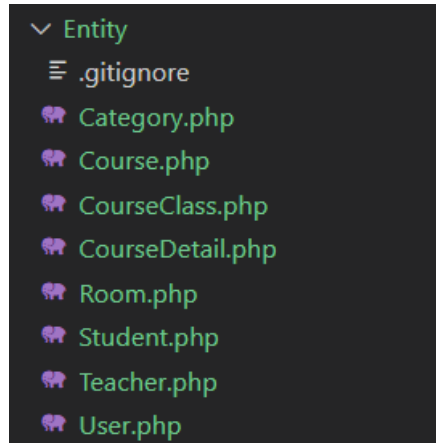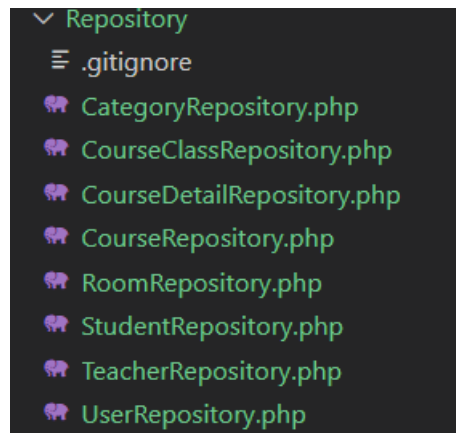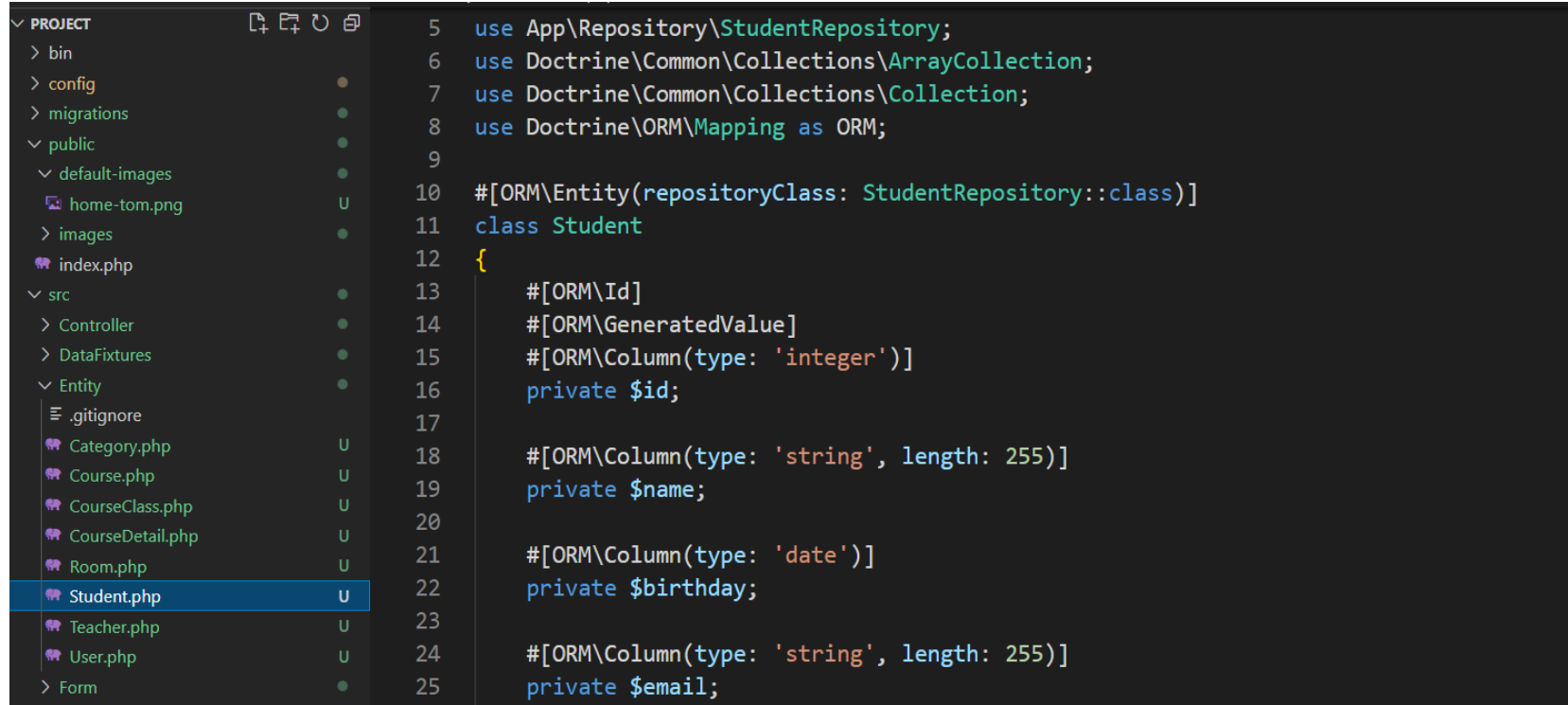

Figure 10.1: Entity folder

Figure 10.2: Repository folder

```php
use App\Repository\StudentRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: StudentRepository::class)]
class Student
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: 'integer')]
    private $id;

    #[ORM\Column(type: 'string', length: 255)]
    private $name;

    #[ORM\Column(type: 'date')]
    private $birthday;

    #[ORM\Column(type: 'string', length: 255)]
    private $email;
```

Figure 11: Student entity

Figure 12: Student orm

2. **Web screenshots**
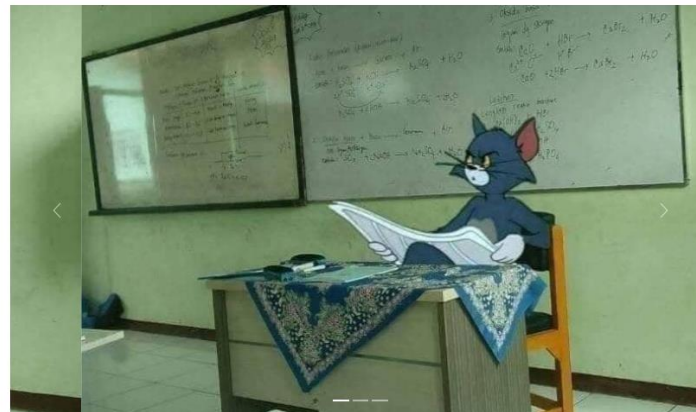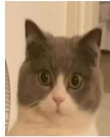
Figure 13: Home and navigation bar interface

Figure 14: View Student

ADD NEW STUDENT

Student Name

Birthday

01 ▾ 01 ▾ 1998 ▾

Email

Phone Number

Student Image

Chọn tệp   Không có tệp nào được chọn

Save

Figure 15: Add student

Figure 16: Update student
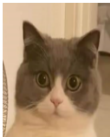
Student ID	Student Image	Student Name	Birthday	Email	Phone Number	Class	Actions
2		da12	29/11/2021	anhnd277@gmail.com	849383830	demo1 demo989 demomax2 demomax3 LOP MAU GIAO	✎ 🗑

Figure 17: Student detail

|	Class Name	Course	Actions
☐ Check all			
☐	demo1	GCH08079	✎
☐	demo12	GCH0807fsdfds	✎
☐	demo45	GCH0806	✎
☐	demo989	GCH0806	✎
☐	demo166666	GCH0807fsdfds	✎
☐	demoMax	ACH0807	✎
☐	demomax2	BCH0807	✎
☐	demomax3	BCH0807	✎
☐	LOP MAU GIAO	GCH0807fsdfds	✎
☐	ljkjlkjl	ACH0807	✎

Delete

Figure 18: View class
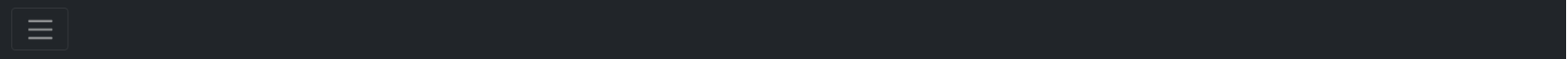
Figure 19: Add class

EDIT CLASS

Class Name

demo1

Maximum Students

30

Start Date

01/12/2021

Start Date

31/12/2021

Course

○GCH0807fsdfds○GCH0806◉GCH08079○GCH0807○ACH0807○BCH0807○GCH08090909

Save

Figure 20: Update class

Class ID	Class Name	Course	Maximum Student	Number Students	Start Date	End Date	Students	Actions
4	demo1	GCH08079	30	3	01/12/2021	31/12/2021	view	✎ 🗑

Figure 21: Class detail

☐Check all	Student ID	Student Name	Student Course
☐	2	da12	demo1 demo989 demomax2 demomax3 LOP MAU GIAO
☐	8	dademo23	demo1 demomax3 LOP MAU GIAO
☐	9	dademo2345	demo1 LOP MAU GIAO
☐	11	da124545fd	

Assign To Class

Figure 22: Assign student to a class

	⊞Asign Student To Class		
☐Check all		Student ID	Student Name
☐		2	da12
☐		8	dademo23
☐		9	dademo2345

Remove

Click to Back

Figure 23: Remove student from a class

V. Conclusion

1. Evaluate

Strength	Weaknesses
- The website has met the minimum requirements of a student management system such as: adding, updating, editing, deleting searching. - The system is built according to the MVC model (Model, View, and Controller). - Website is easy to use, and all functions can be performed by no more than four operations. - In the process of running the website, there are no errors, the functions work very efficiently. - The website has authorization, access denial, and password encryption. - Easy to modify and improve in the code.	- Low security. - There are no special functions other than simple functions such as add, edit and delete. - Simple and boring interface. - Users other than admin can only view. - Some tables in the database are not connected properly. - Some features are not logical.

Figure 24: Pros and Cons

2. Future Improvement

In the future, the system will be added with more features such as grading function, student attendance, assignment submission, etc. In addition, the team will improve the user interface of the web to make them look better and more interactive. In the near future, the team will try to correct the shortcomings and irrationalities in the system to build a complete website.

VI. Appendix

Group member list	Role
Nguyen Duc Anh (GCH17051)	User Login, Student and Class function back-end and front-end
Pham Khue	Room and Teacher function back-end and front-end
Nghiem Cao Nhan	Course Category and Course function back-end and front-end

Figure 25: Member list

Link GitHub: https://github.com/paulpham98/projectWeb