

**Undergraduate Final Year Project Report**

**PROJECT REPORT**

**Development of eTutoring web-based secure role-based system for Greenwich University**

**Group 3**

**Phan Tuan Khoa (GCS17397), Truong Duc Long (GCS16560), Phan Quang Duc, Ngo Nguyen Song Nguyen, Vo Doan Phuong Duy**

**Bachelor of Science with Honours in Computing**

**TABLE OF CONTENTS**

[**1.** **Introduction** 2](#_Toc39688148)

[**1.1. Project aim & objectives** 3](#_Toc39688149)

[1.1.1. Analyze the system 3](#_Toc39688150)

[1.1.2. Design the system 3](#_Toc39688151)

[1.1.3. Implementation 4](#_Toc39688152)

[1.1.4. Testing the system 4](#_Toc39688153)

[**1.2. Legal, Social, Ethical and Professional** 4](#_Toc39688154)

[1.2.1. Legal 4](#_Toc39688155)

[1.2.2. Social 5](#_Toc39688156)

[1.2.3. Ethical 5](#_Toc39688157)

[1.2.4. Professional 5](#_Toc39688158)

[**1.3. Planning** 5](#_Toc39688159)

[**2. Database** 6](#_Toc39688160)

[**3. Functionality** 7](#_Toc39688161)

[3.1. Login 7](#_Toc39688162)

[3.2. Admin add new user 8](#_Toc39688163)

[3.3. Admin delete user 9](#_Toc39688164)

[3.4. Admin/Staff add new class 10](#_Toc39688165)

[3.5. Admin/Staff delete class 11](#_Toc39688166)

[3.6. Admin/Staff Add Student, Teacher with Class in Group 12](#_Toc39688167)

[3.7. Admin/Staff delete Group 13](#_Toc39688168)

[3.8. Teacher/Student upload file 14](#_Toc39688169)

[**The Activity of website coding:** 15](#_Toc39688170)

[**4. Testing** 22](#_Toc39688171)

[4.1. Test case 22](#_Toc39688172)

[4.2. Test result 25](#_Toc39688173)

[**5. Agile Method** 30](#_Toc39688174)

[**TABLE OF FIGURES** 32](#_Toc39688175)

[**TABLE OF REFERENCES** 32](#_Toc39688176)

# **Introduction**

Greenwich University wants to develop a web-based, role-based eTutoring system to aid students in learning more effectively. After some discussions internally, they want to implement Scrum with Agile concepts in this project and choose us to be the scrum team that will handle this project.

We are a small group of students tasked with building the system for Greenwich University. In our scrum team, we have:

**Long** – the ScrumMaster. He works with documents and finalize the report.

**Khoa** – works together with the ScrumMaster and other team members to handle documents.

**Duc** – the database designer. He is in charge of designing the database and connect it with the front-end.

**Duy** – the programmer and tester who works together with Duc to handle the back-end. He has some professional experience in working as a professional software developer.

**Nguyen** – the web designer who handles the front-end. He has some professional experience in working as a professional software developer.

In this project, we will develop the system using PHP and MySQL for the back-end. Both of them are free and open-source technologies that will significantly reduce the development cost, comparing to Microsoft SQL with its expensive pricing. Combining with XAMPP which acts as a local host or a local server, we can quickly test the system and reduce the development cost further. For the front-end, we use the common HTML, CSS and JavaScript. All of which are popular technologies with tons of supports from their communities. And we decided to use Lavarel for its easy database migration, simple and clean code that makes it easy to maintain, and being a free open-source framework. We especially love the integrated security measures that helps prevent attacks such as SQL injection and CSR forgery.

In this system, there will be three types of users: Students, Tutors, and Staffs. **The Students** can use the system for meetings with their tutors, upload their documents and commenting on them, blogging, and have their own personal dashboard that summarize their interactions.

**The Tutors** will also be able to use the system for meetings with their students, uploading documents and commenting on them and blogging. The interesting thing is that they will have a dashboard of their personal students. It can be sorted and filtered the way they want.

**The Staffs** have the highest authorities. They can allocate and reallocate personal tutors to students and both the students and tutors will receive notification emails when this happens. The allocation can be done in bulk of ten at a time. Certain authorized staffs can gain access to dashboards of other staff as well as individual dashboards for students.

# **1.1. Project aim & objectives**

With this project, we want to learn how to develop an eTutoring system as a team using Scrum with Agile concepts in 6 months.

In order to achieve our aim, we will define the objectives and describe them in details below.

## 1.1.1. Analyze the system

We start with defining the requirements and functions of the system based on the specification given to us. By defining them, we understand how the system works and decide on the appropriate approach for this project. The requirements of the system as well as technologies and methodologies chosen by us will affect our plans for the project. This shouldn’t take more than two weeks.

The deliverables for this objective would be the backlog, where all of the requirements and tasks to develop the finished product are listed and prioritized.

## 1.1.2. Design the system

After analyzing the system and planning out the project, we start designing the system. By designing the system, we can code according to our design. This can take from two to four weeks.

The deliverables for this objective would be wireframes for the front-end, ERDs and Activity Diagrams for back-end.

## 1.1.3. Implementation

After making the design, we will start coding accordingly to create a functional system with working functions. The system will be finished before the deadline arrives.

The deliverables for this objective would be the source code and screencasts to show the system in action.

## 1.1.4. Testing the system

It is important to test the system that we develop to ensure that they are working as intended. Within every sprint, we test the functions that we are developing.

The deliverables for this objective would be a test plan, a test description, and a test report.

# **1.2. Legal, Social, Ethical and Professional**

## 1.2.1. Legal

PHP is a free open-source scripting language under the PHP License. The project is in accordance to the mentioned license.

MySQL is an open-source relational database management system. The project is in accordance to the MySQL Commercial License.

XAMPP is a compilation of free software that is free of charge and free to copy under the terms of the GNU General Public License. The project is in accordance to the mentioned license.

The Lavarel framework is an open-sources software licensed under the MIT License. The project is in accordance to the mentioned license.

JavaScript is a programming language licensed under the GNU General Public License. The project is in accordance to the mentioned license.

HTML 5 is a markup language licensed under the W3C License. The project is in accordance to the mentioned license.

This project belongs to our team and University of Greenwich. In this project, every materials that are used here will come from free sources or legal copies.

## 1.2.2. Social

The system will be used for the benefits of the University of Greenwich, the tutors and students.

Measures will be taken to ensure the security and privacy of the parties involved.

## 1.2.3. Ethical

This project is the work of our team and other involved parties. There will be no plagiarism in this project.

We ensure the privacy of the parties involved and prevent unauthorized access as well as inappropriate use of data.

We will put in our best effort to prevent cybercrimes that can be committed using our system.

## 1.2.4. Professional

Confidential information will not be disclosed or use for personal gain or a third party’s benefit without permission from the University of Greenwich.

We will seek and accept honest criticism about our work, as well as respect and value alternative perspectives to continuously improve ourselves.

We ensure that the code that we write is tested thoroughly and meets our quality standards. We will also continuously improve the code to keep up the quality.

# **1.3. Planning**

Backlog

# **2. Database**



Figure 1 ERD

# **3. Functionality**

## 3.1. Login



## 3.2. Admin add new user



## 3.3. Admin delete user



## 3.4. Admin/Staff add new class



## 3.5. Admin/Staff delete class



## 3.6. Admin/Staff Add Student, Teacher with Class in Group



## 3.7. Admin/Staff delete Group

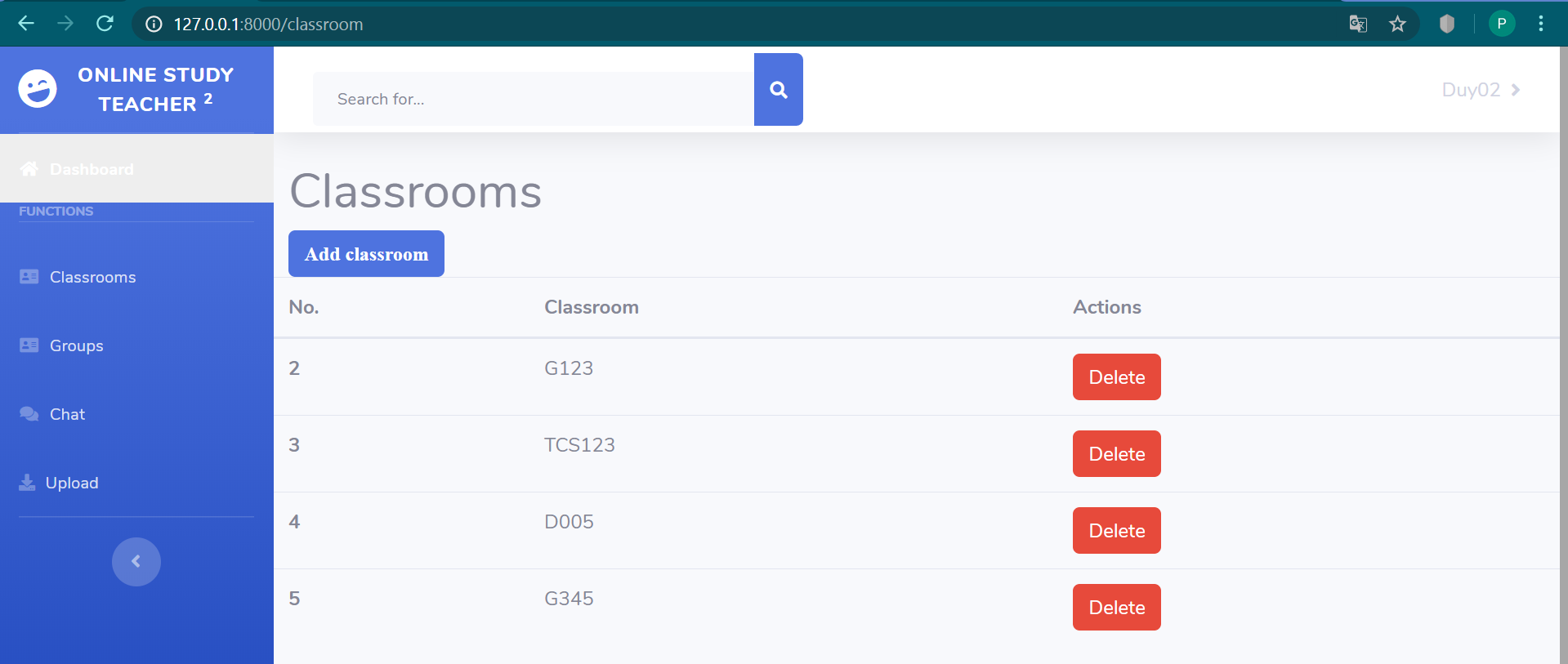


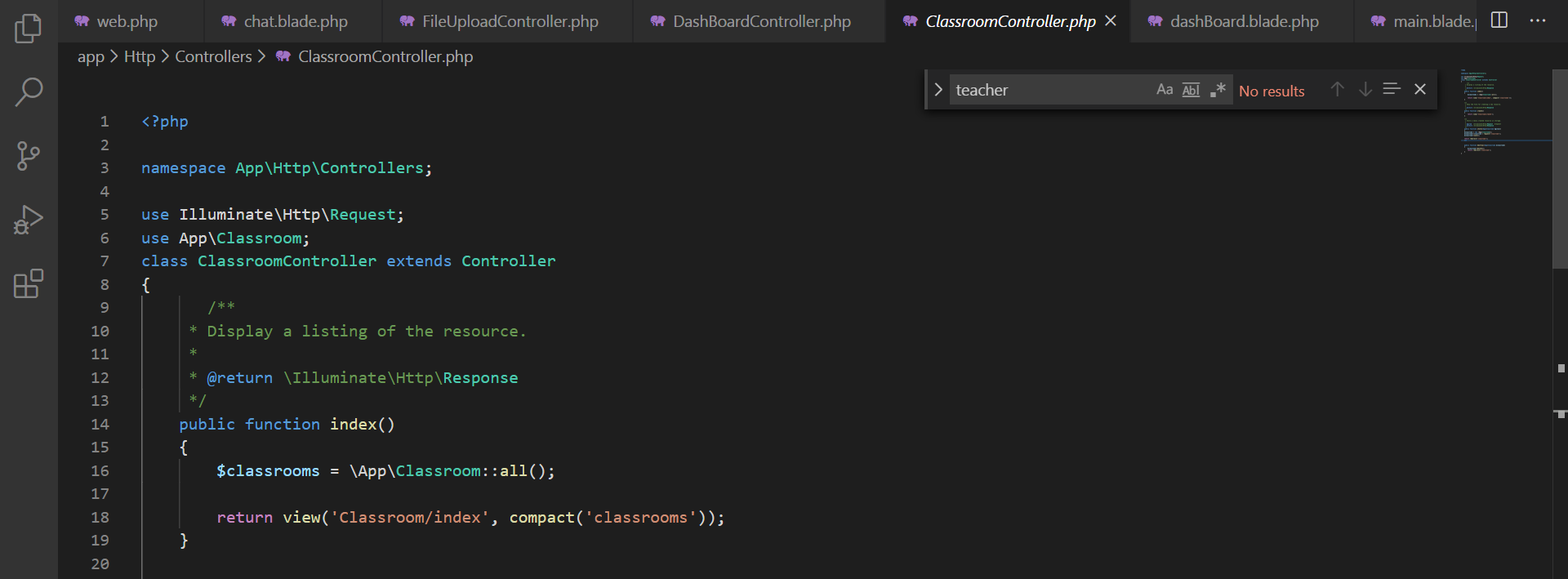
## 3.8. Teacher/Student upload file



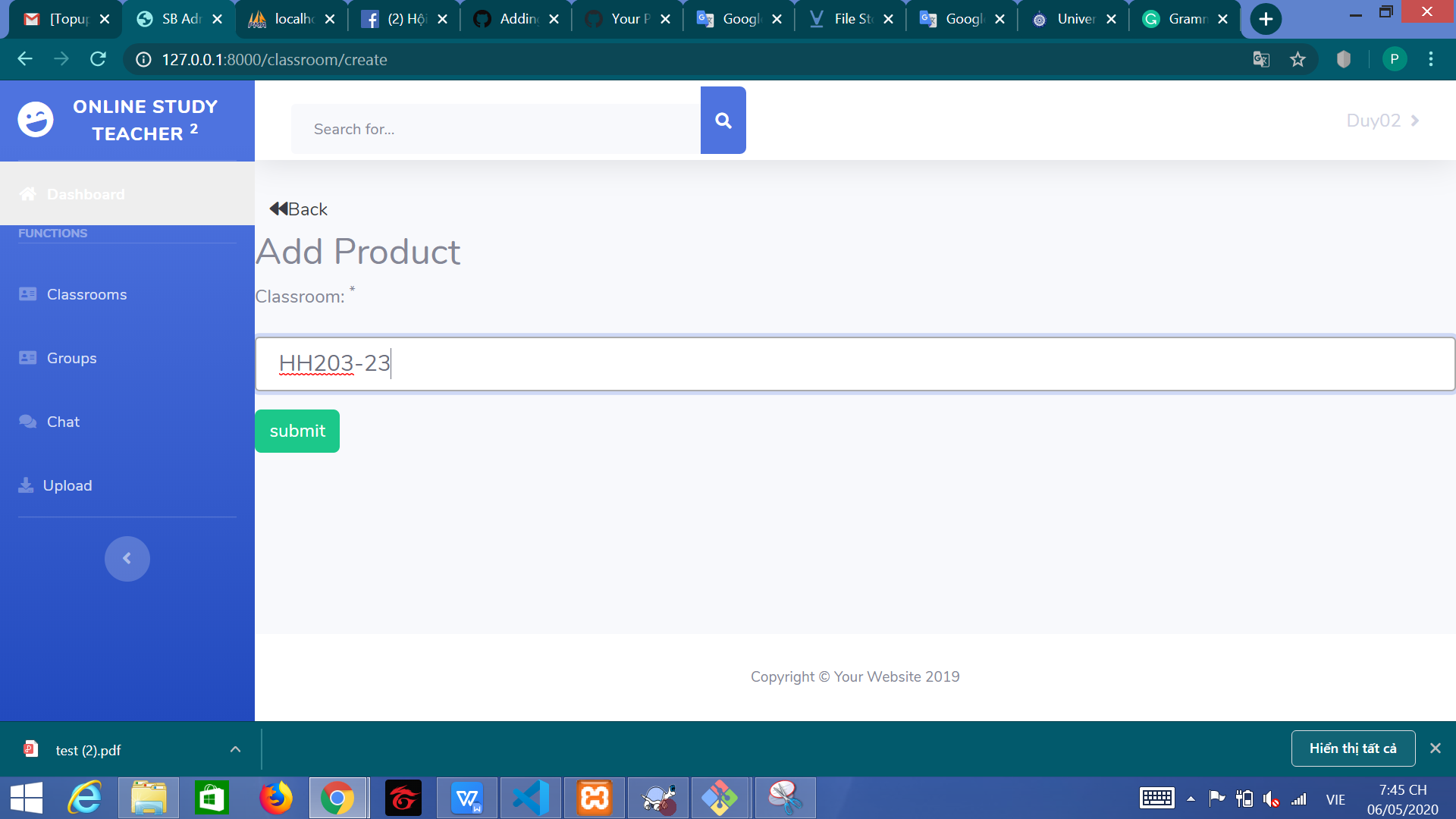
# **The Activity of website coding:**

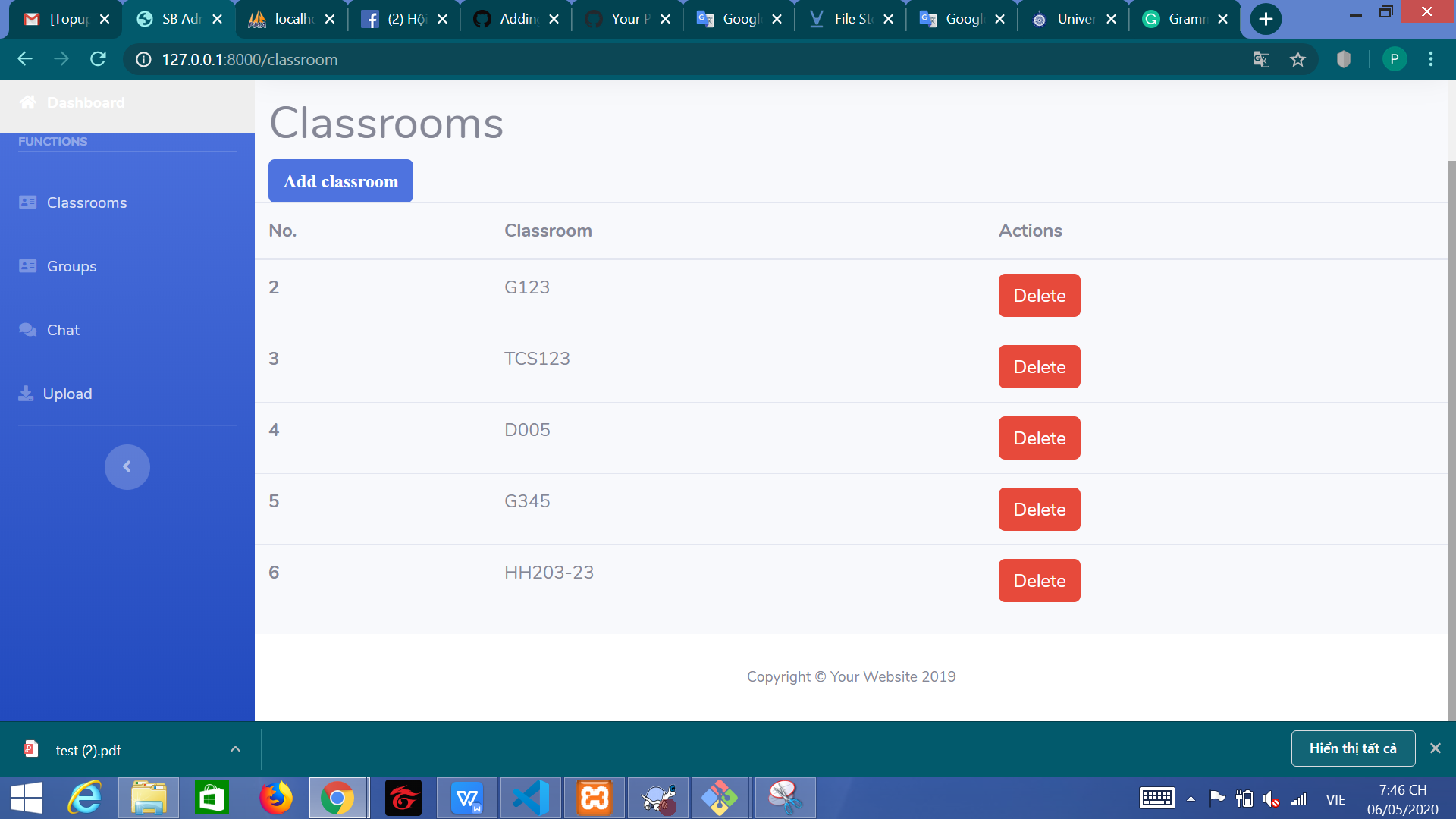
1. Add Classrooms

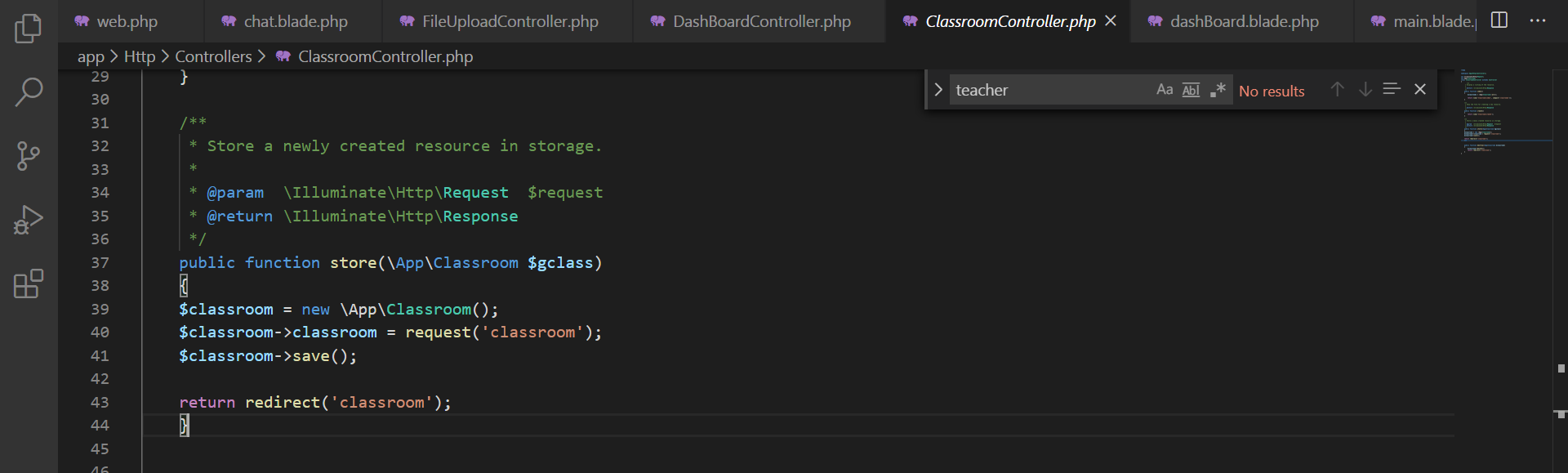




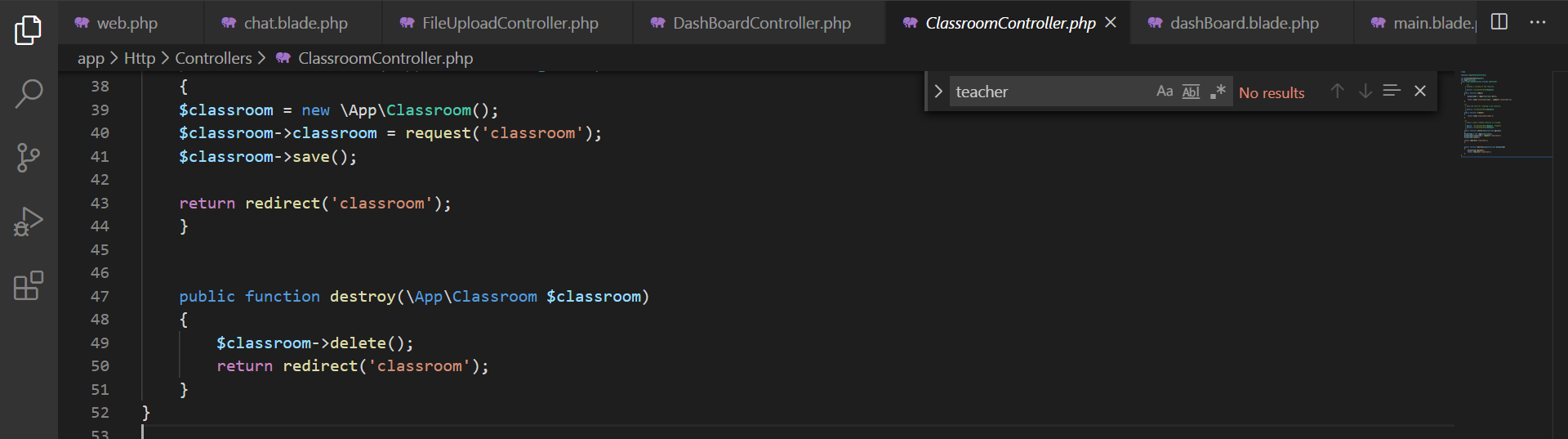
This function is get data in database and show classroom lists.

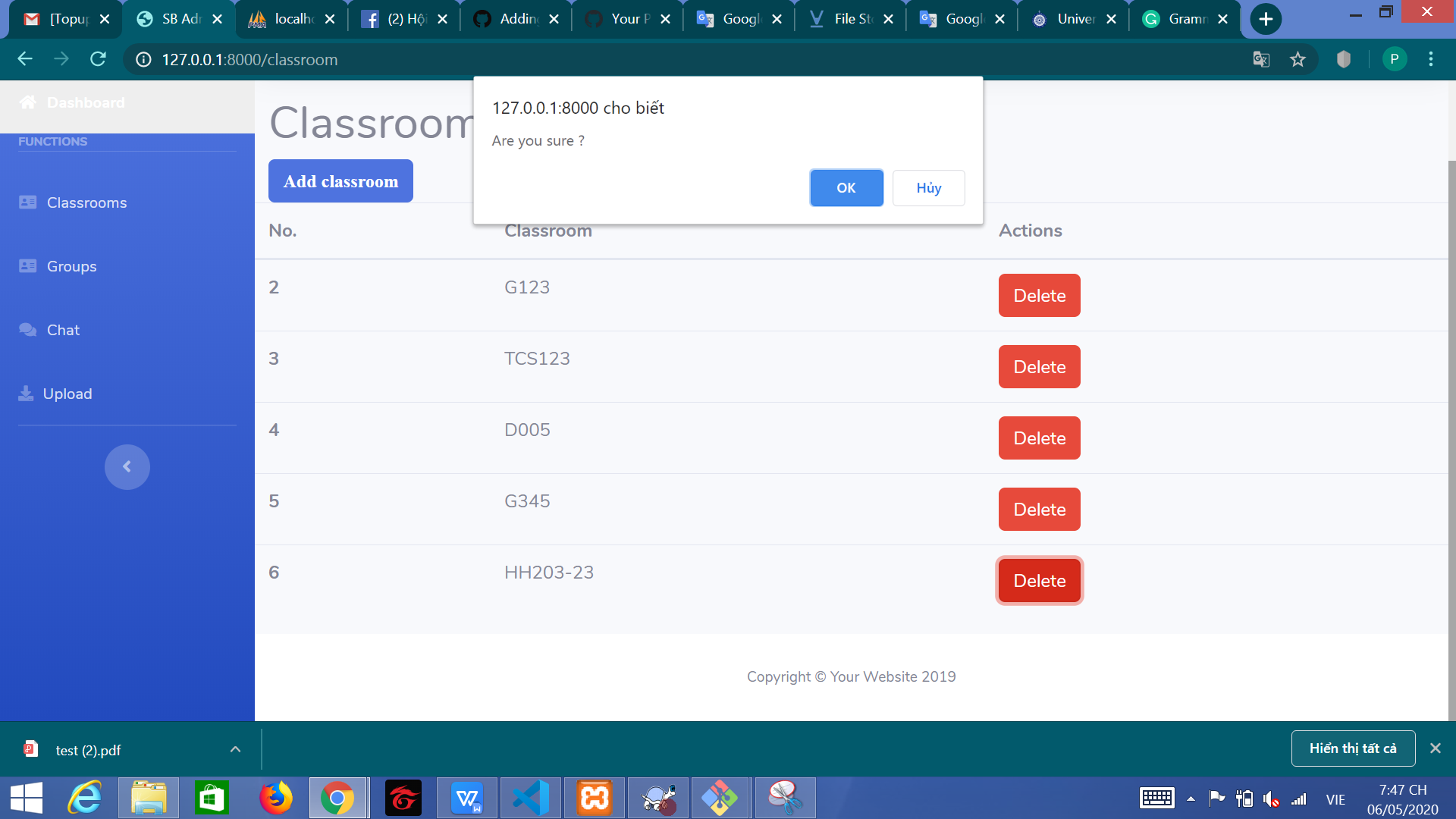


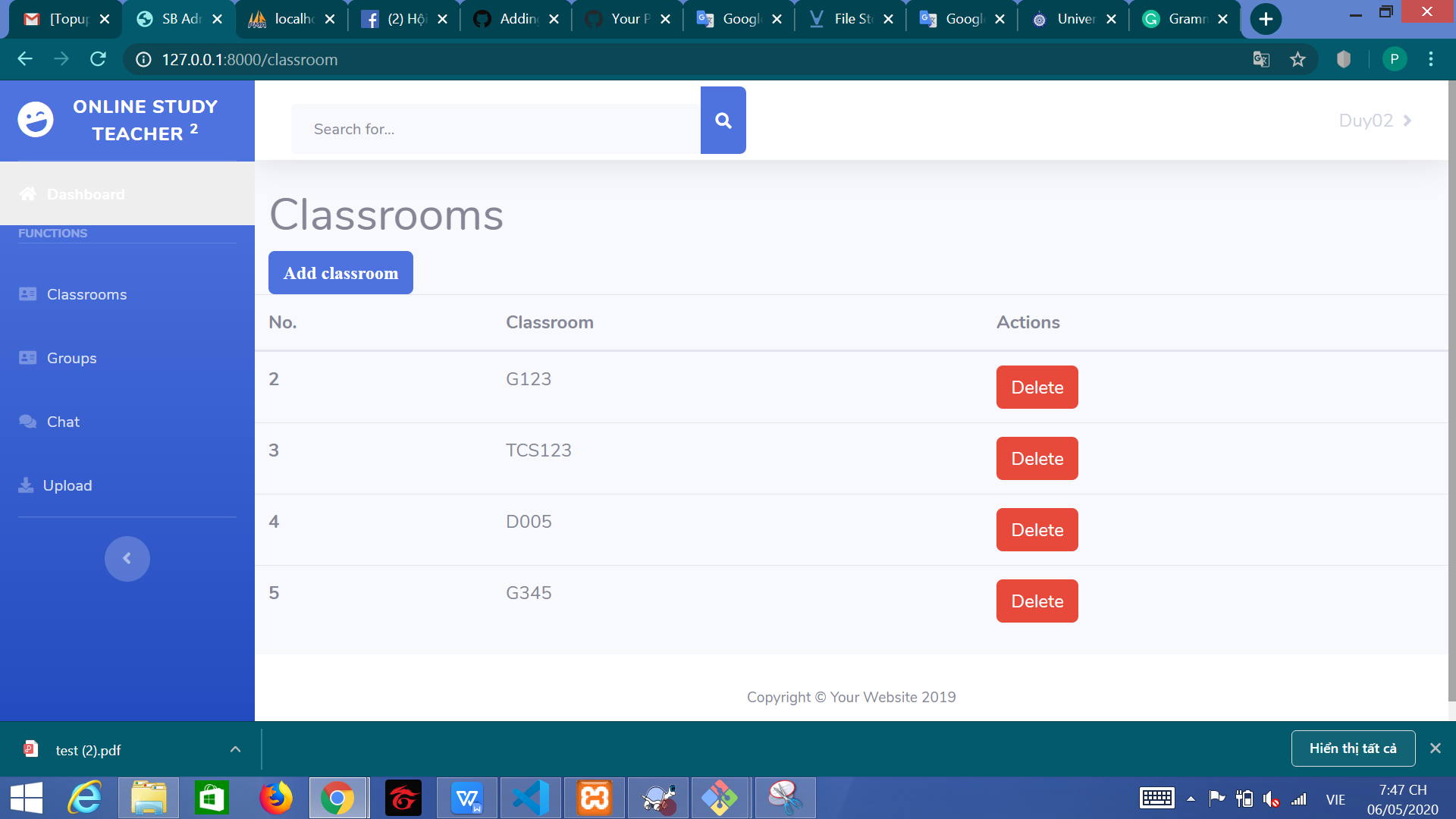




This is function to create new classroom

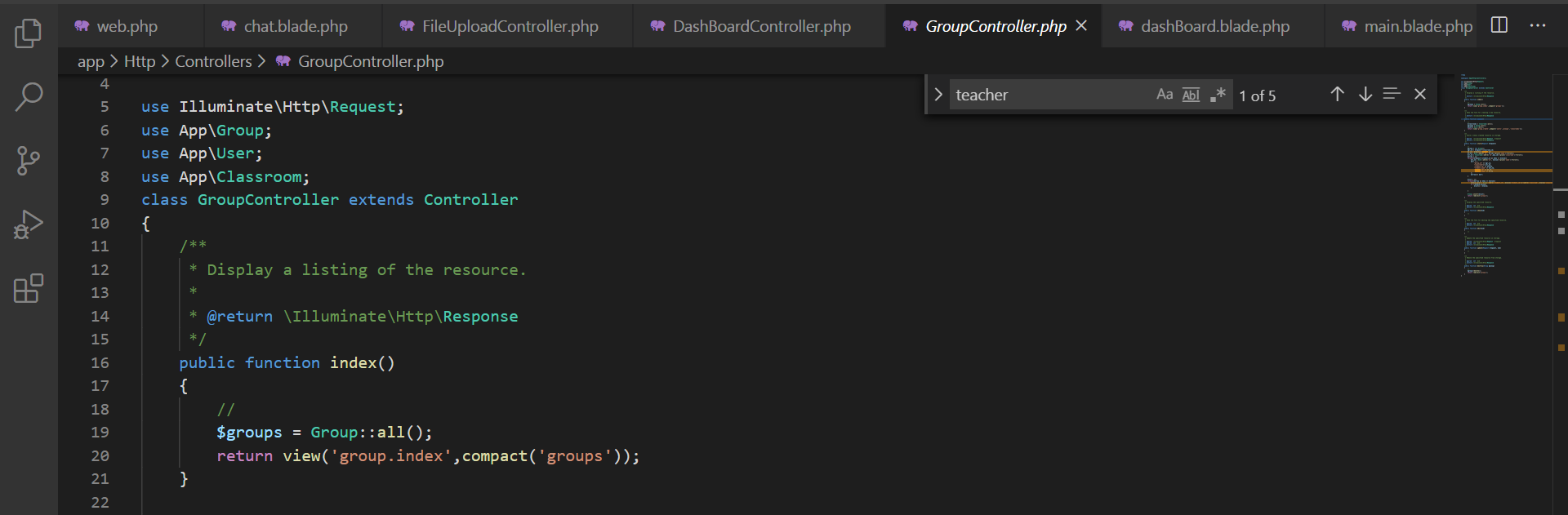






This function have delete classrooms

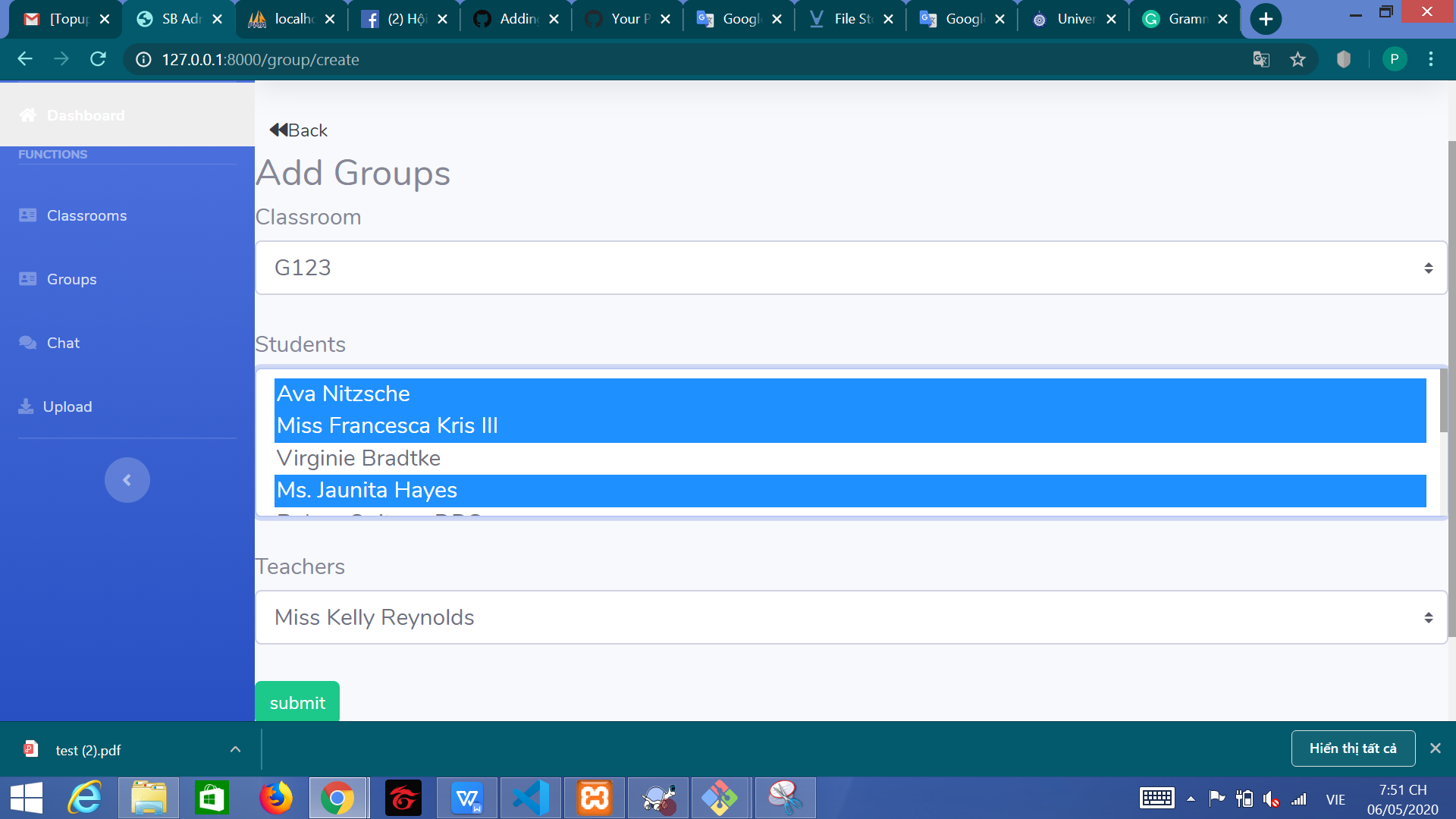
1. Groups



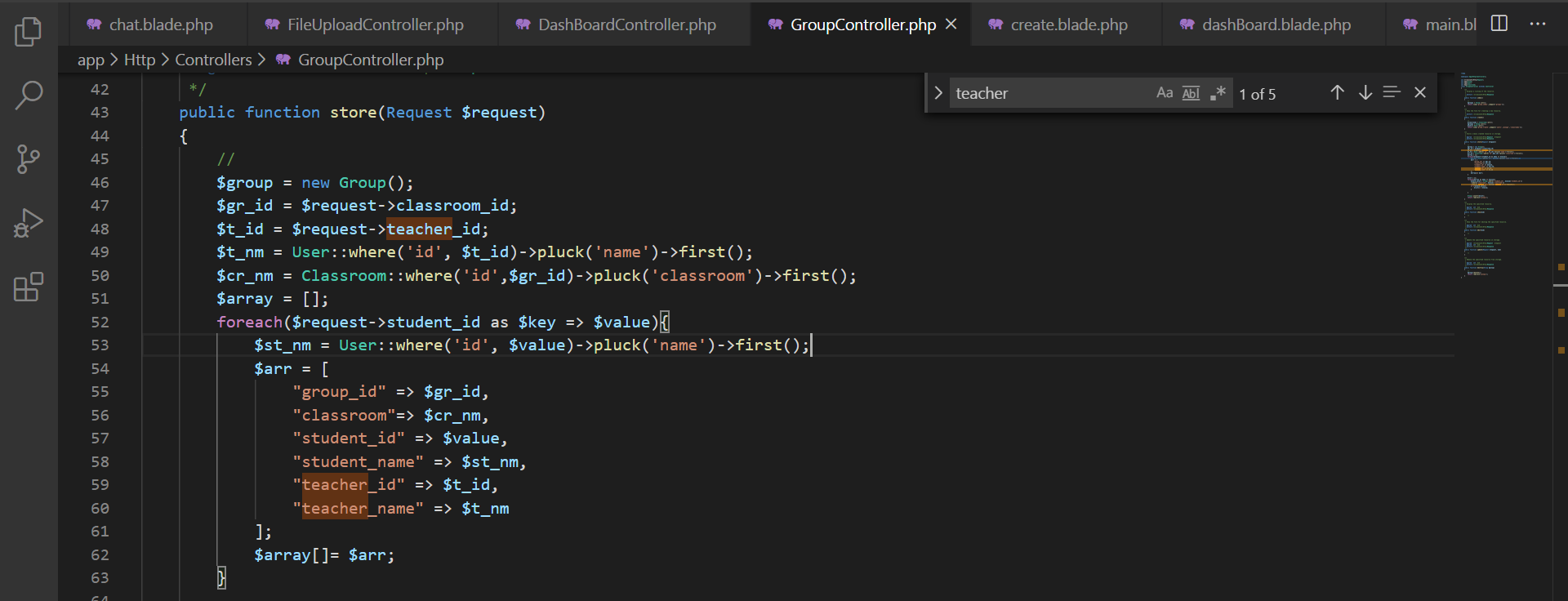


This function have show groups list





This function is Add Groups. It have a special things in this function. You can add multiple users. You can keep Ctrl in keyboard and click all users you want to add. Then this is my code and i have explain this code

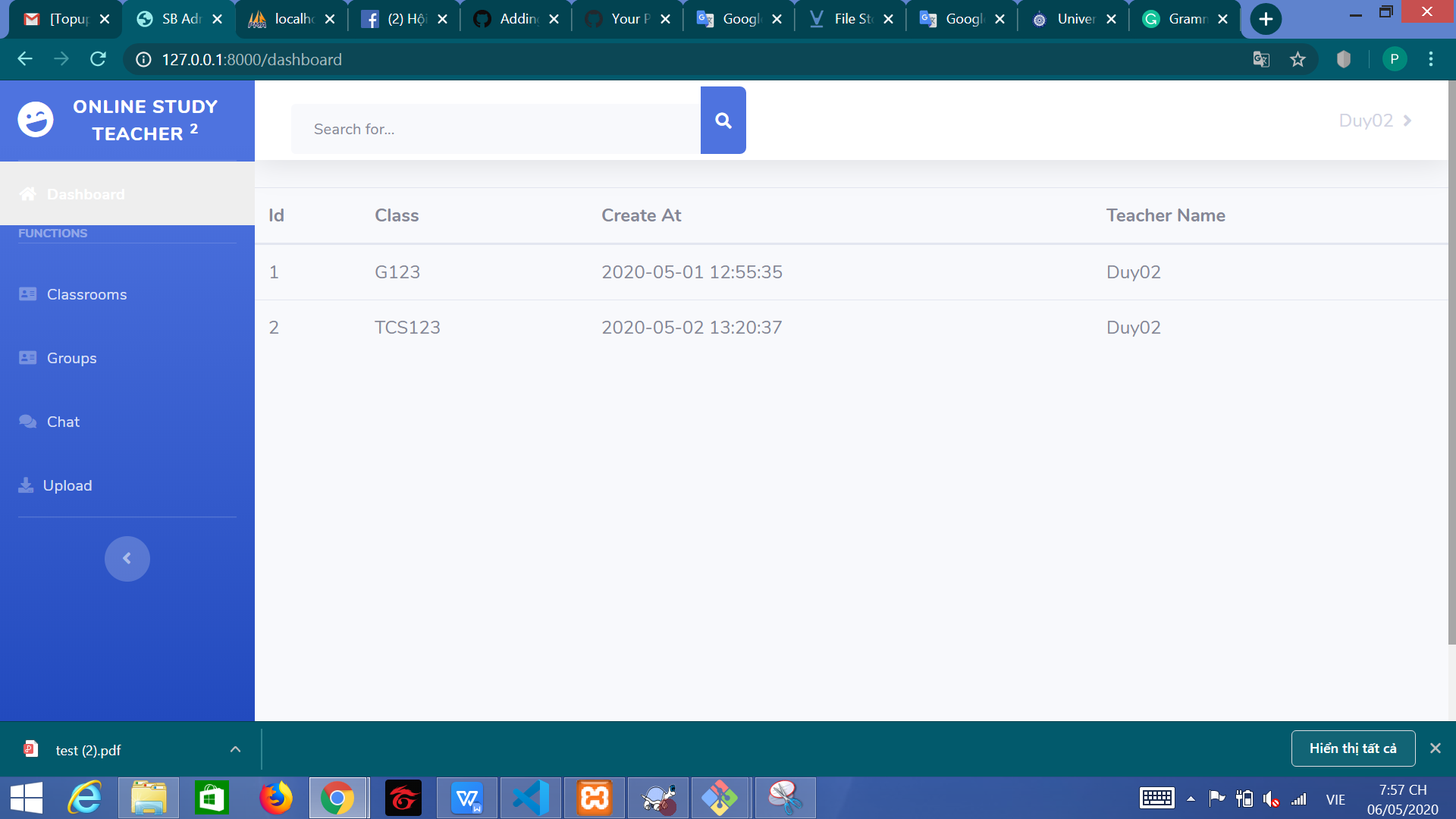


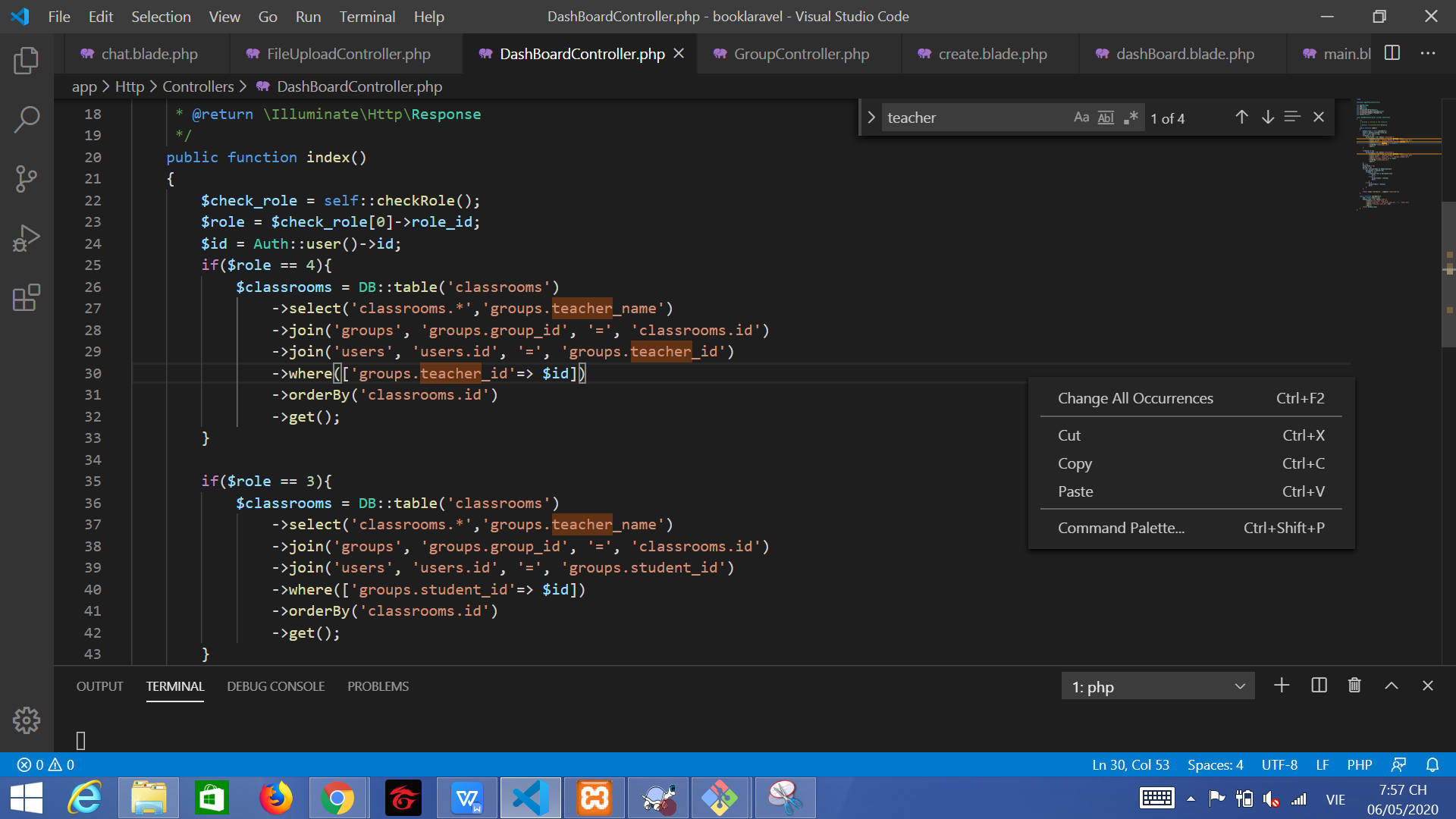
I have using foreach to insert multiple data. But If you choose user, you have add before. Don’t worry.



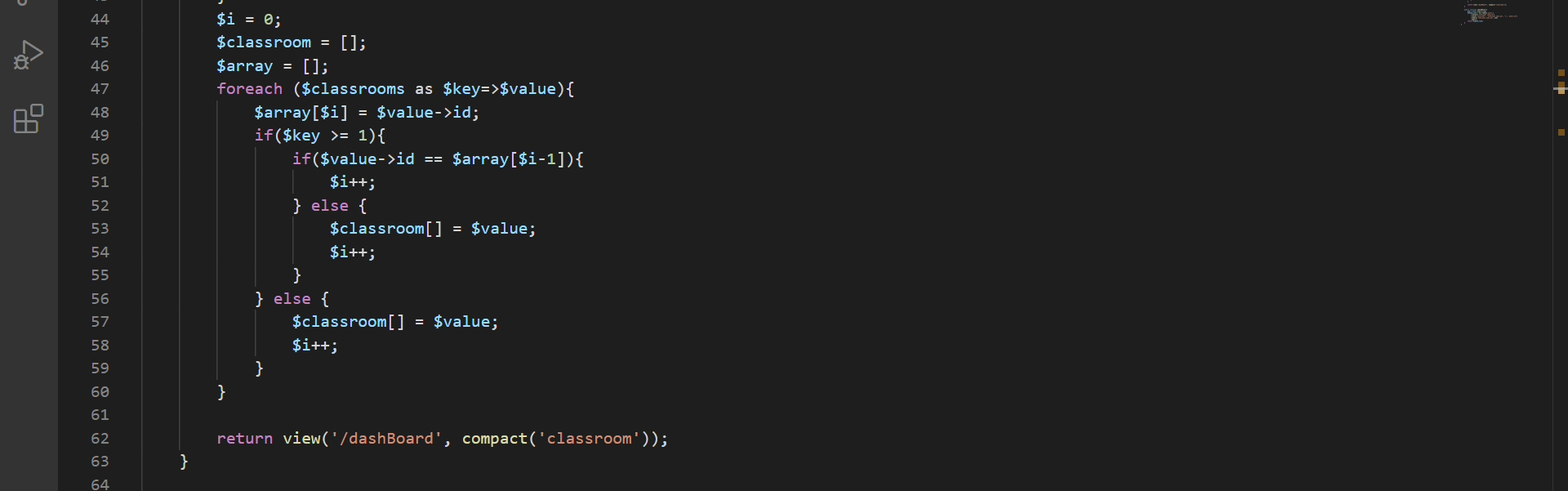
I have to check if user have add in database. This foreach have remove this users and not save in database.

1. Dashboard



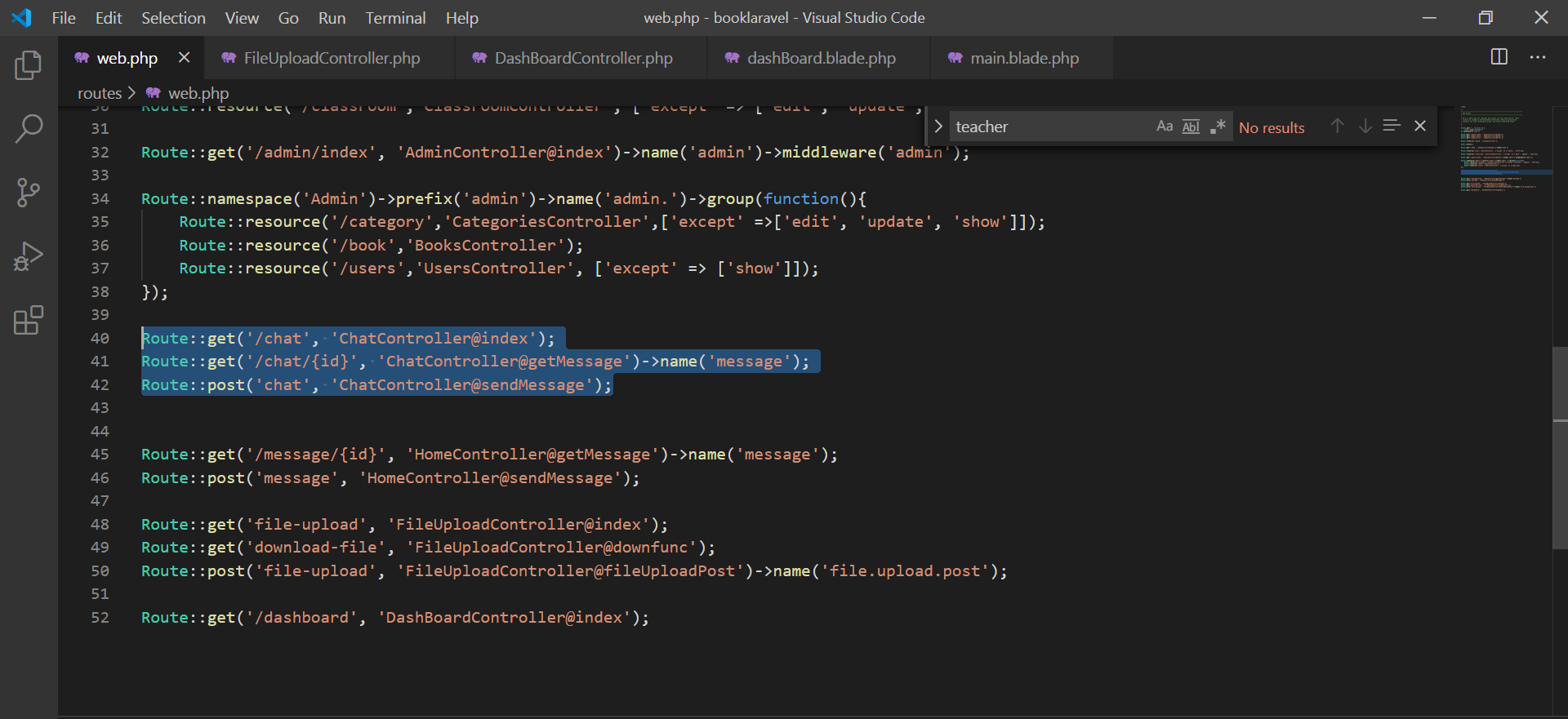


This function have get all class user have joins. But in teacher role, i have get same value a lot. I need to check the same data. This is the code check

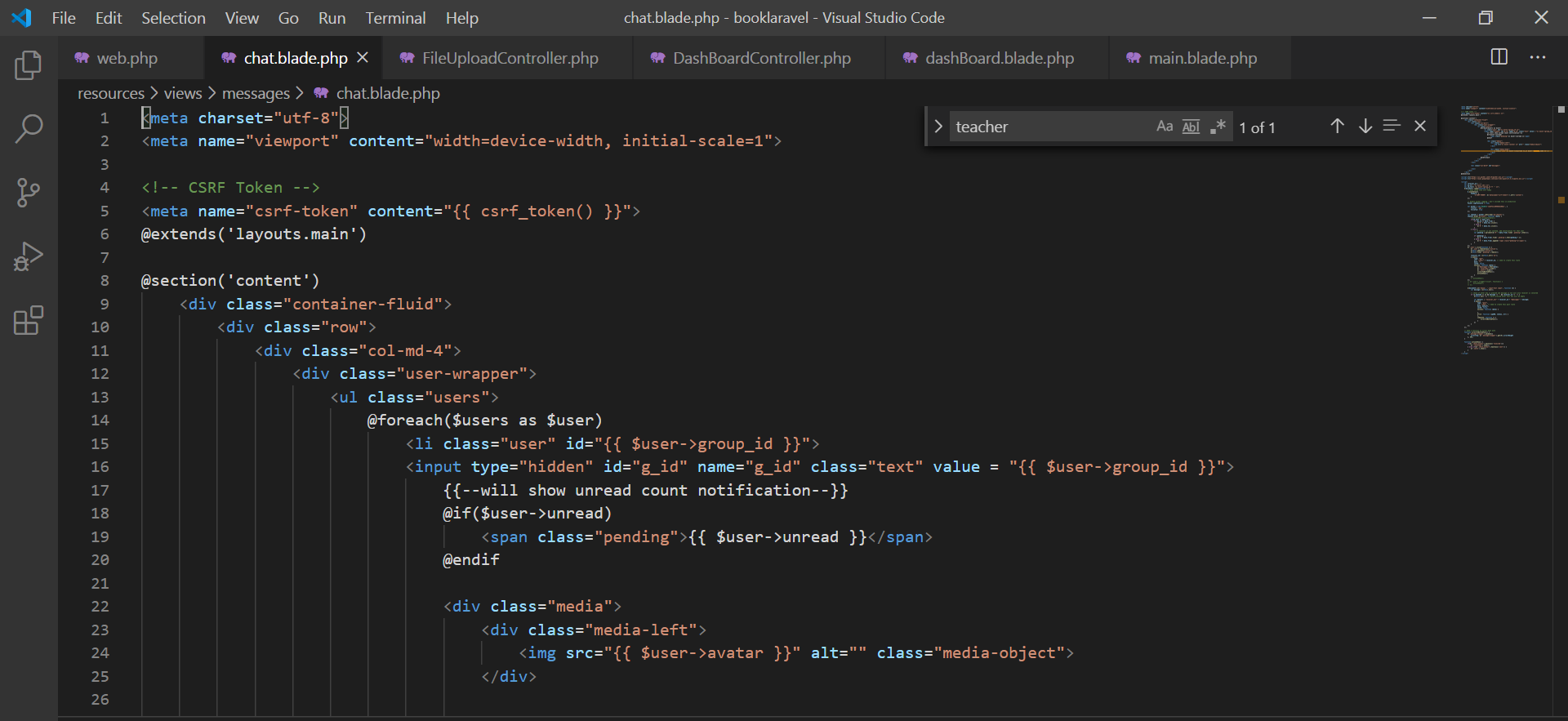


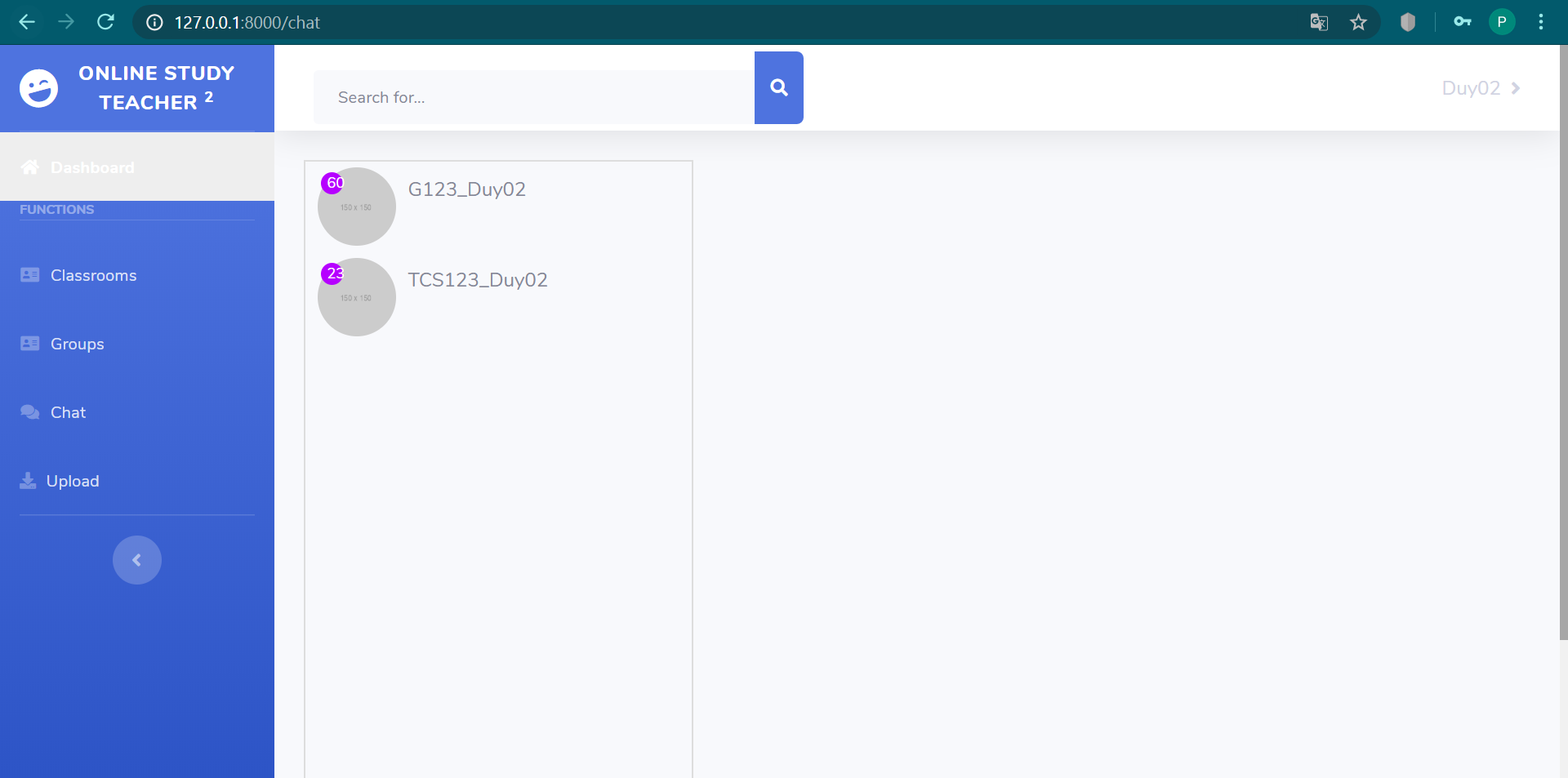
I have using the next variable in array and check before this variable.

1. Function Chat Groups

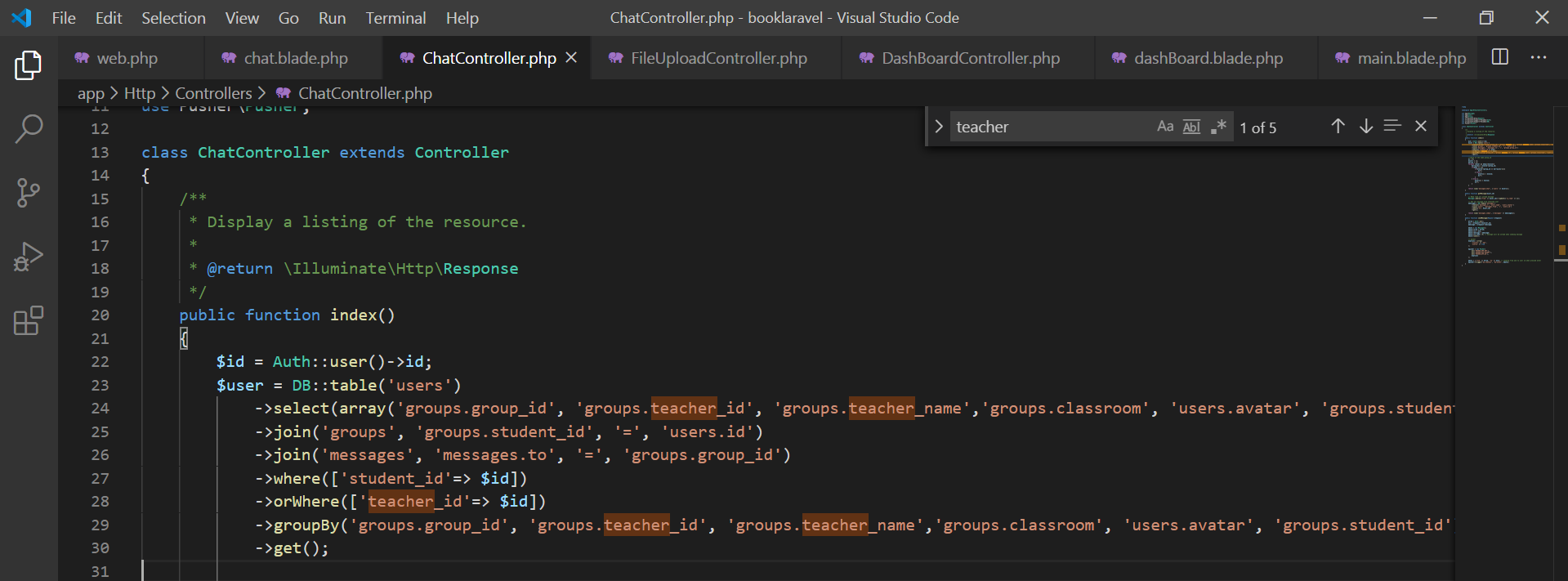


This is a route using in function Chat Groups

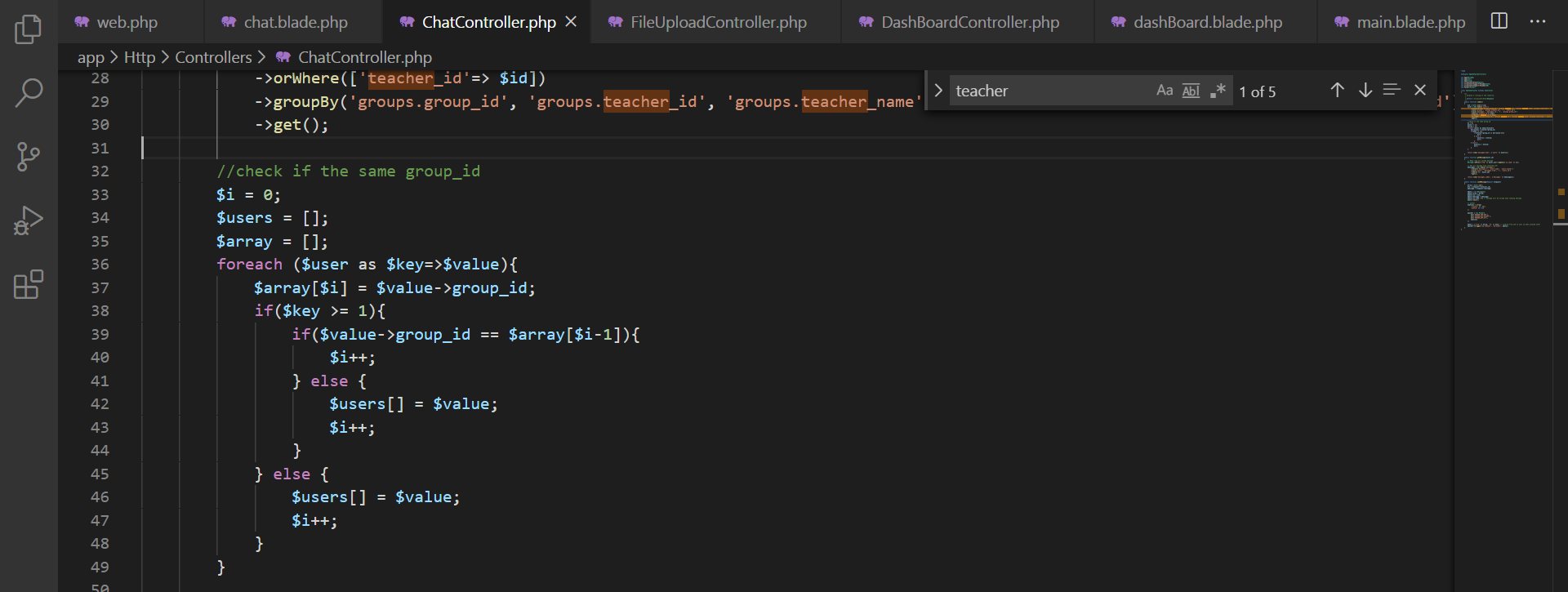




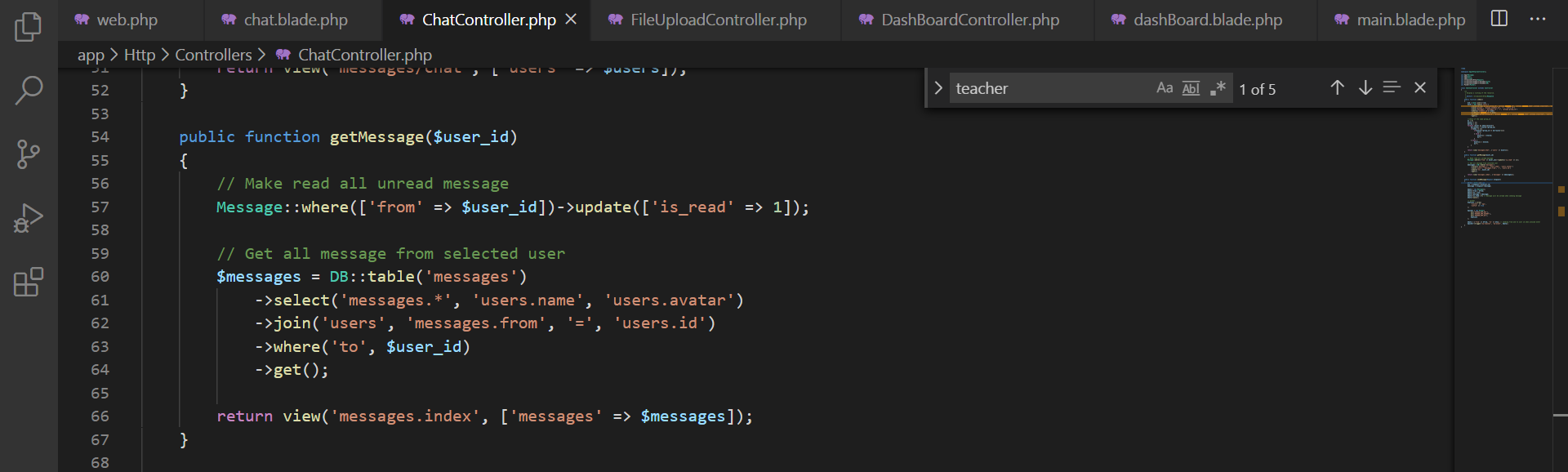
This is HTML show list groups you can have join in class

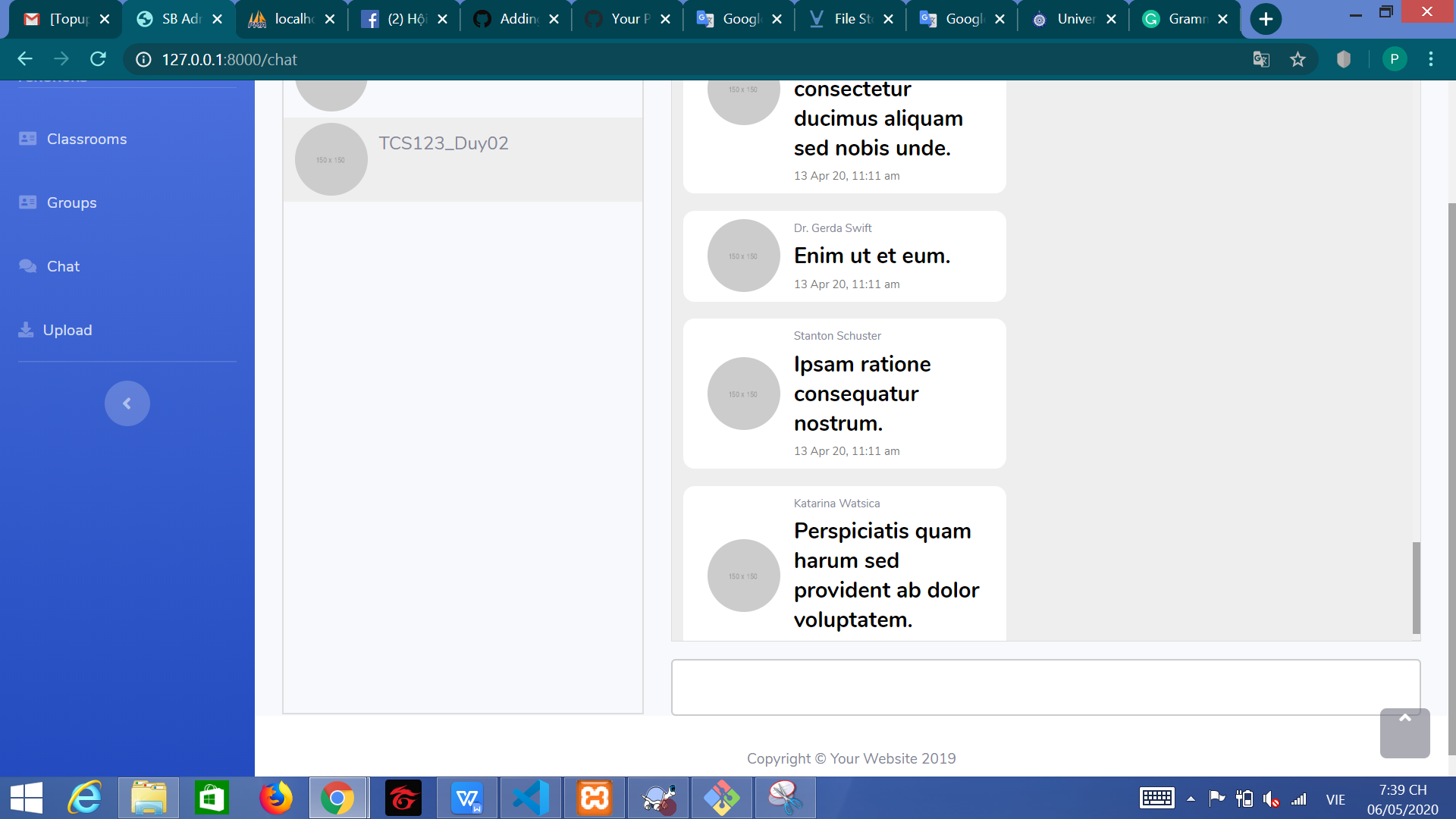


This is query select list Groups

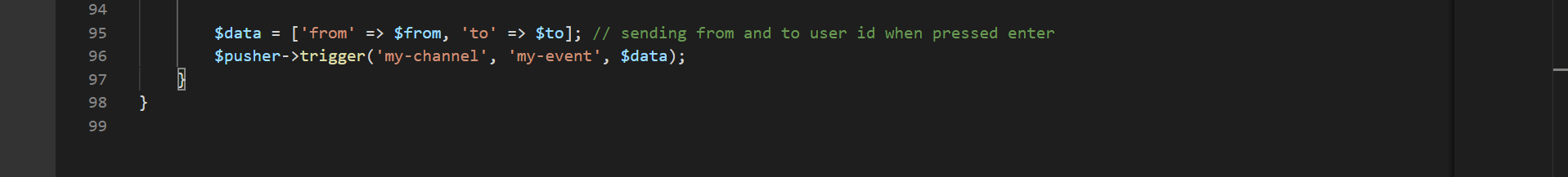
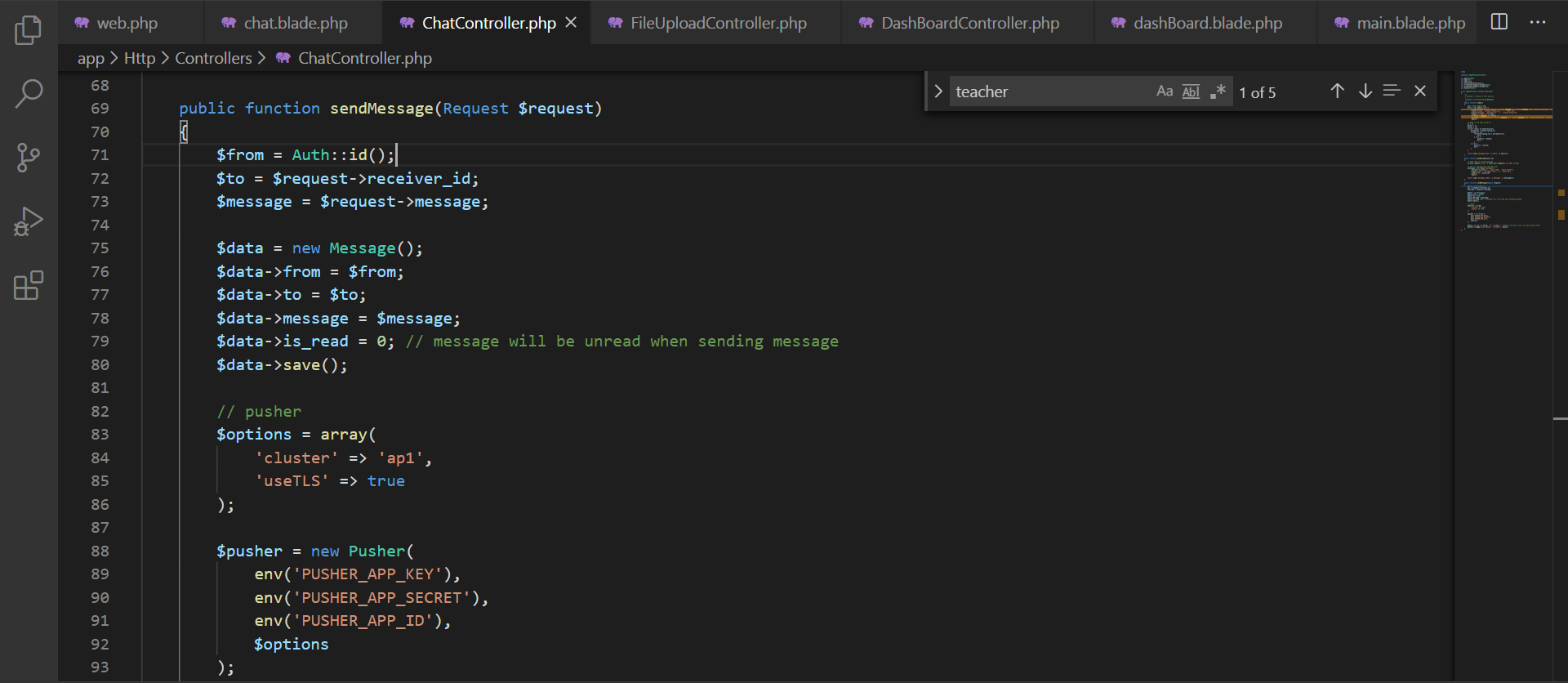


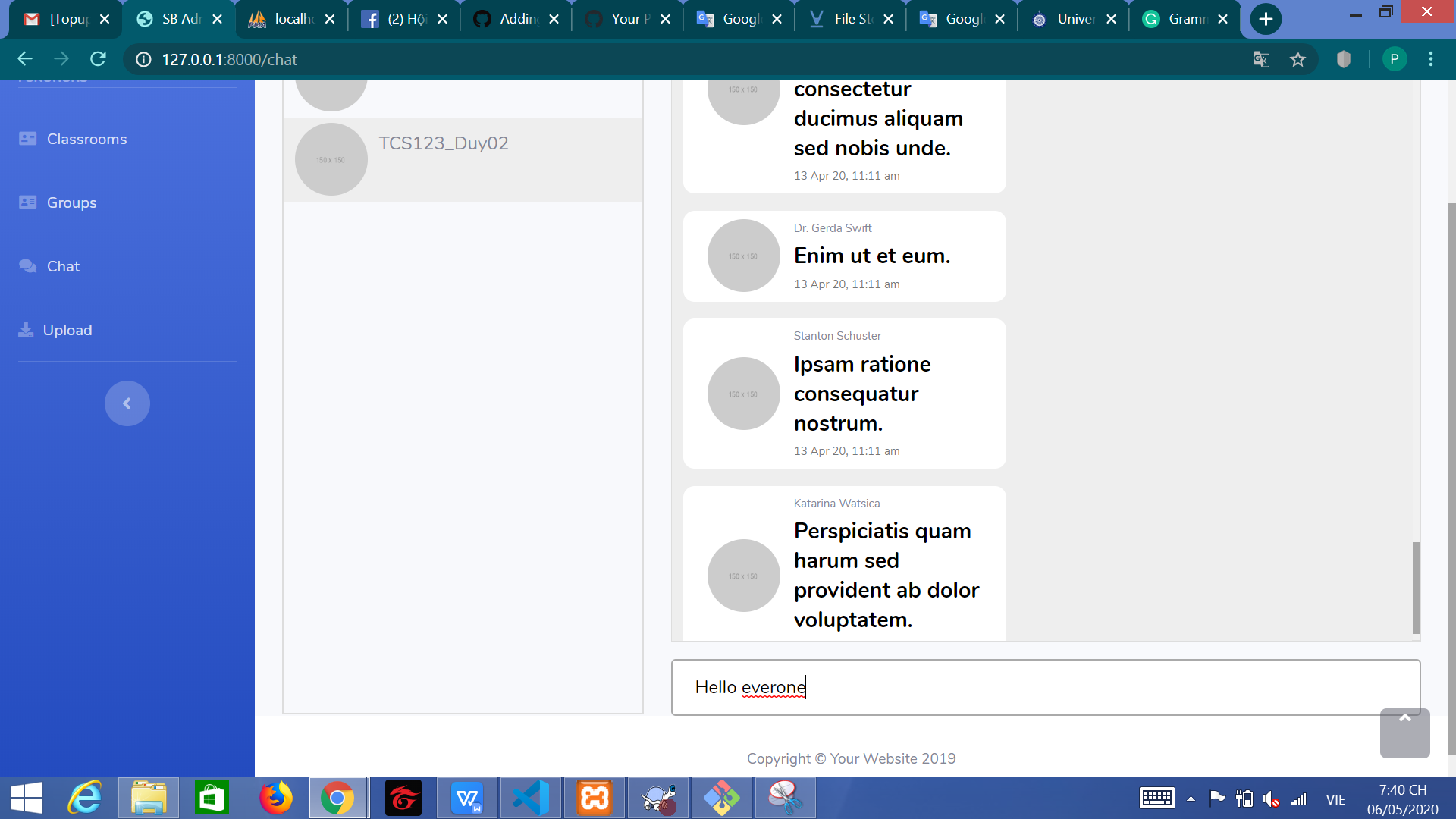
This is speacial code. This code know check the same value after they get value in query builder. This code i have check group\_id before group\_id in array

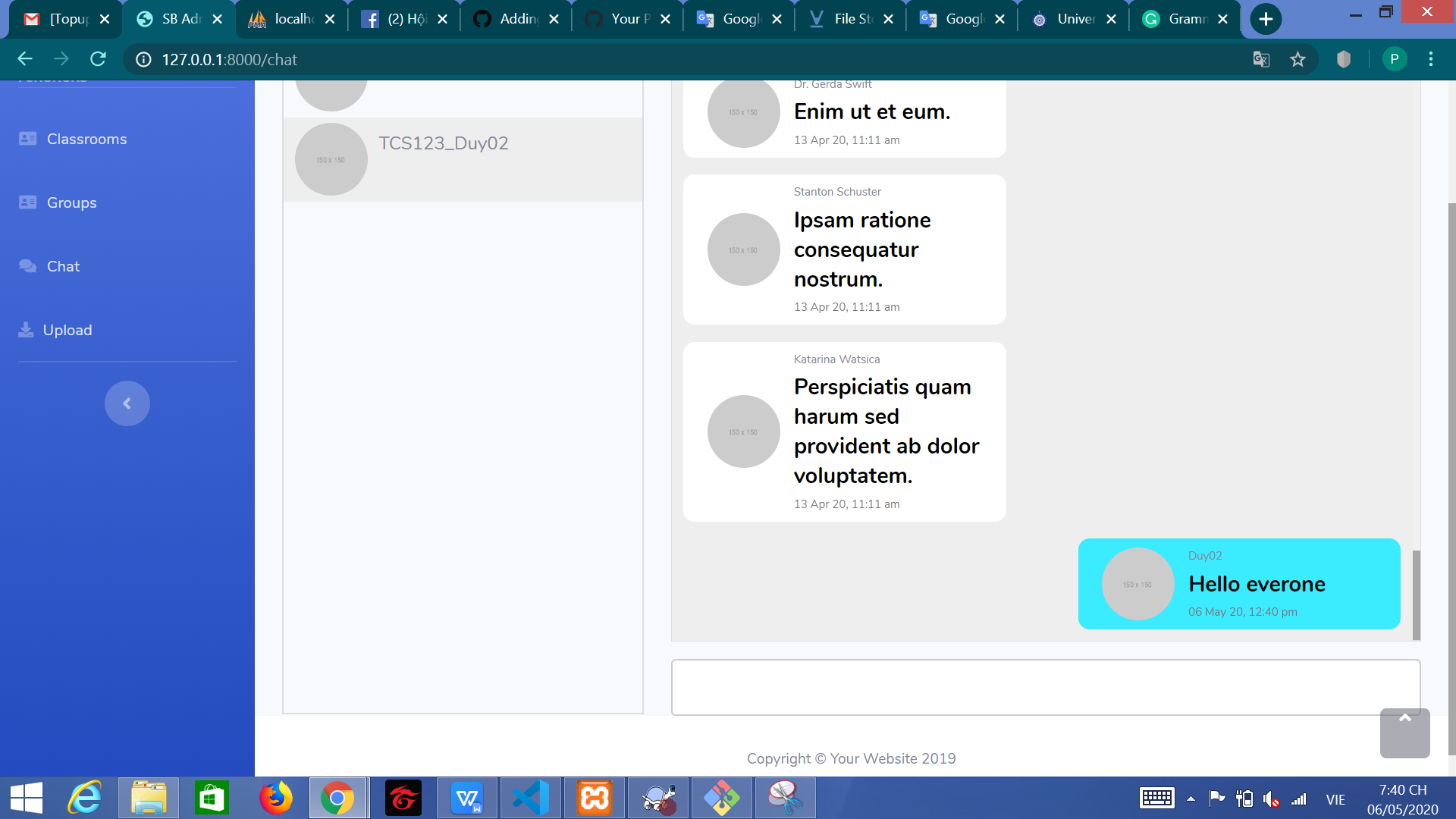




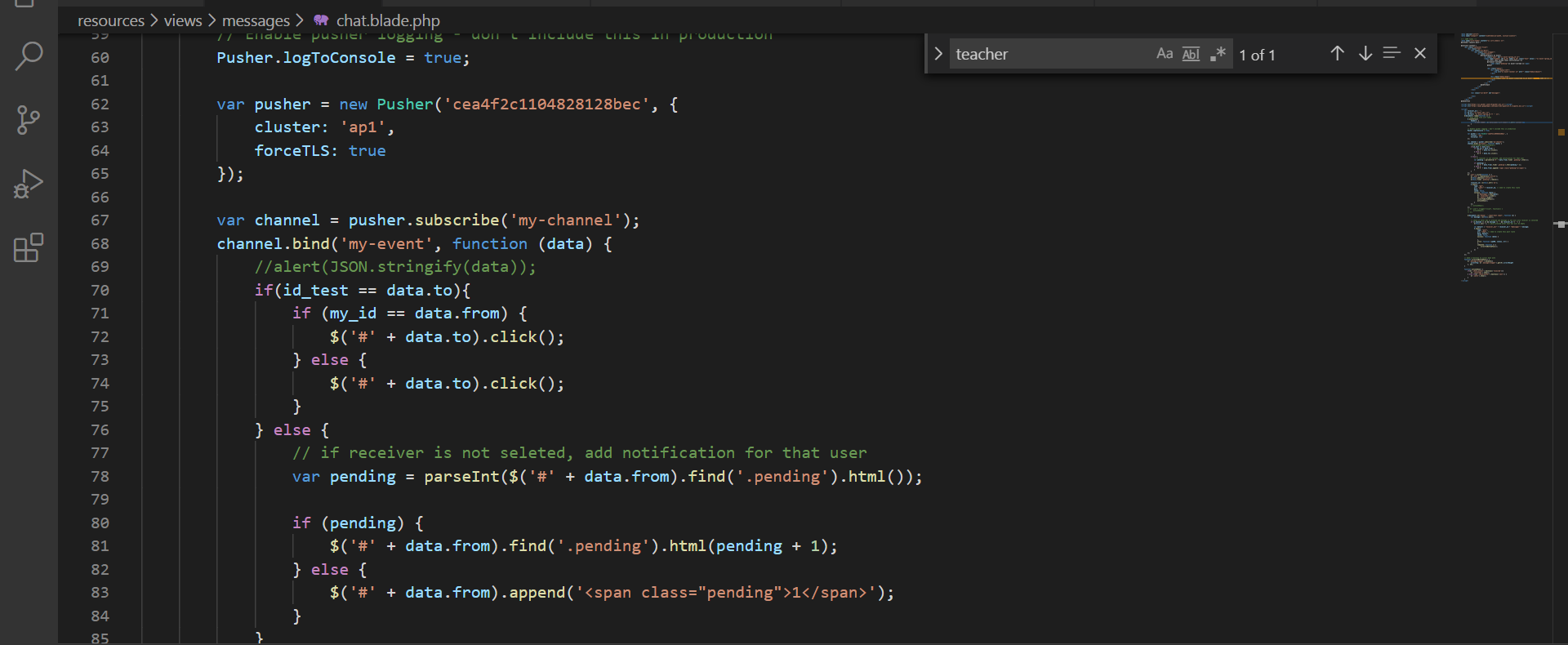
This code get message and show screen if you click chat box





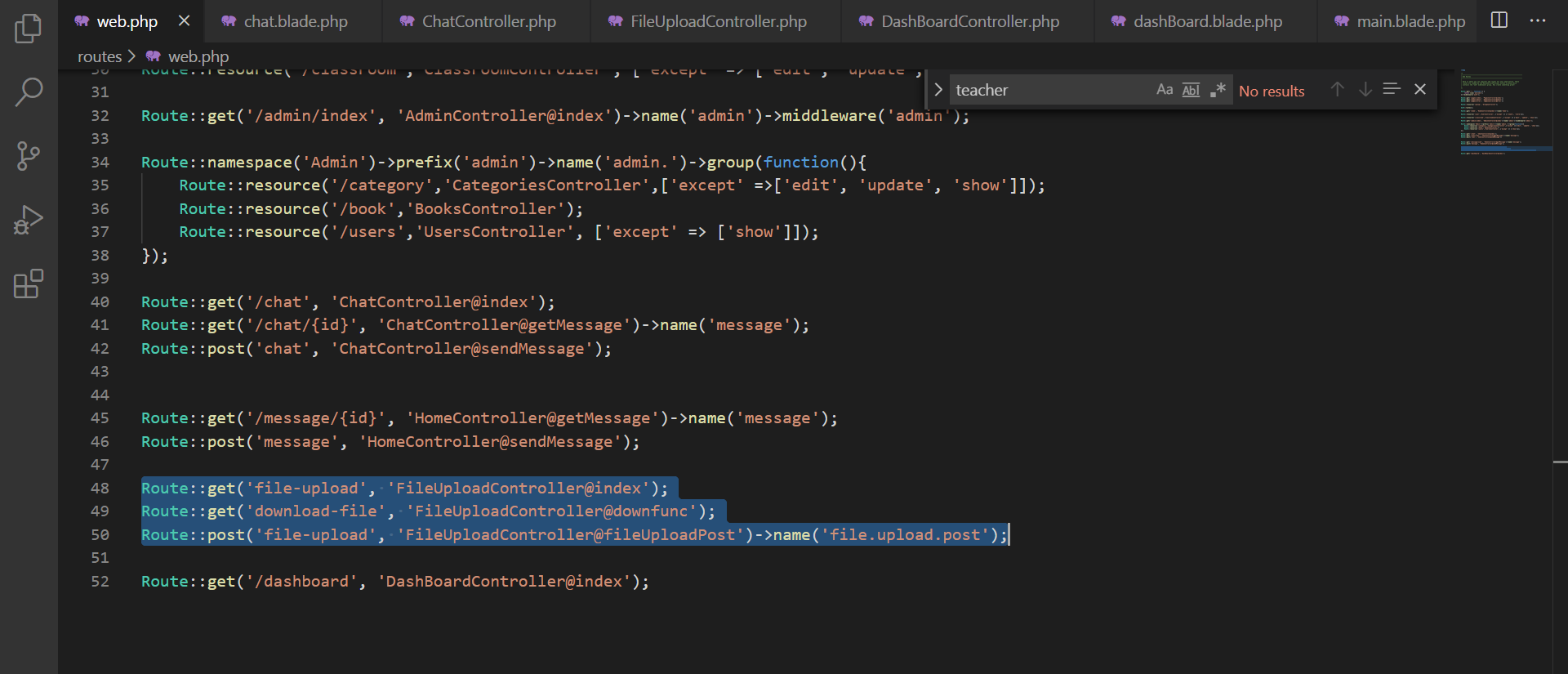


This code have send message. I using Pusher to call API and send message

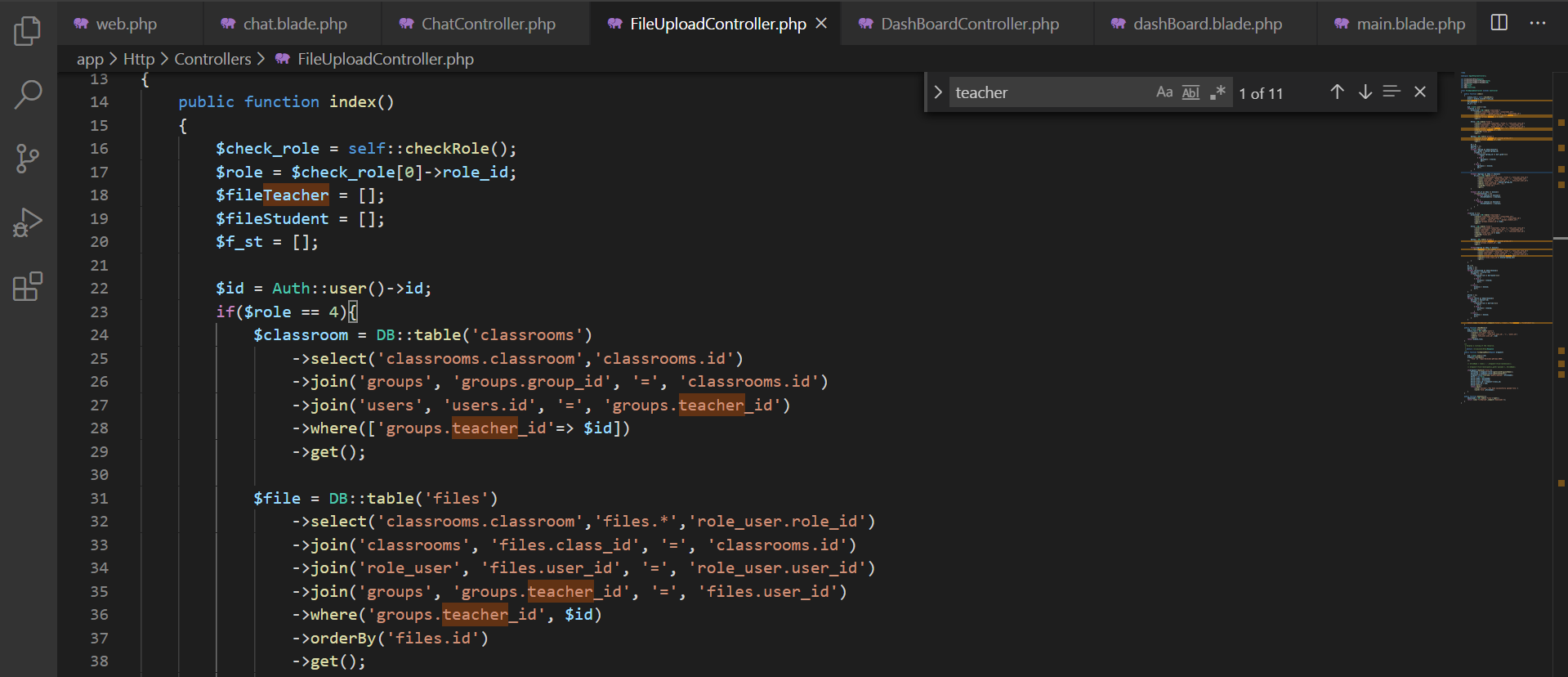


This code have show message when you sent message. I using Jquery in this stages. And this code will show a new messages

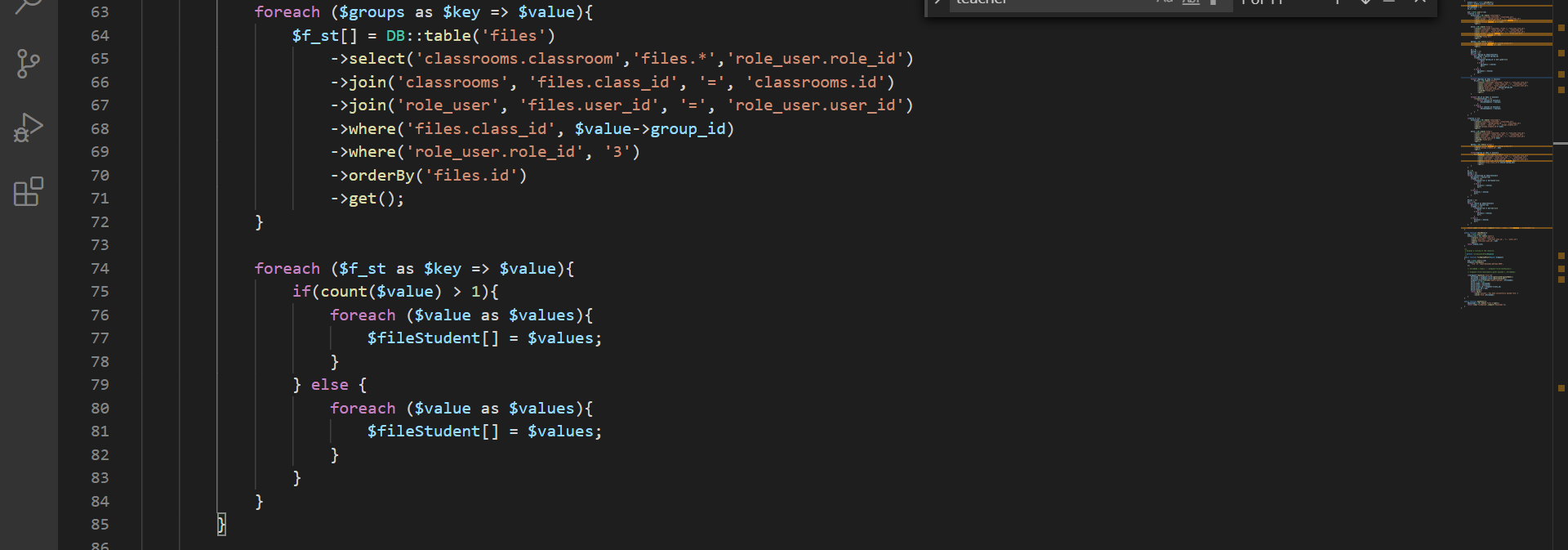
1. Upload Files

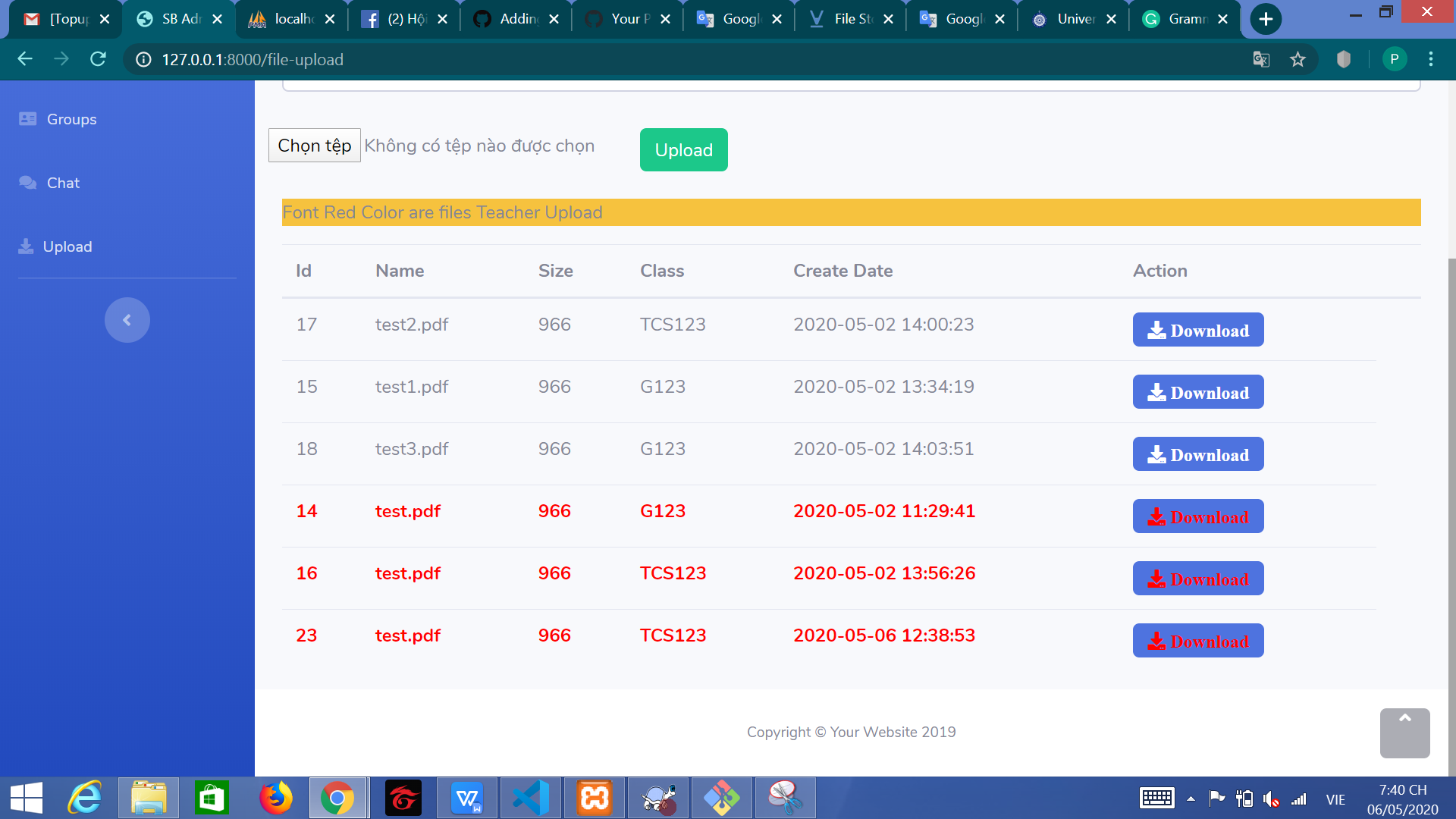


This is route of function Upload FIles

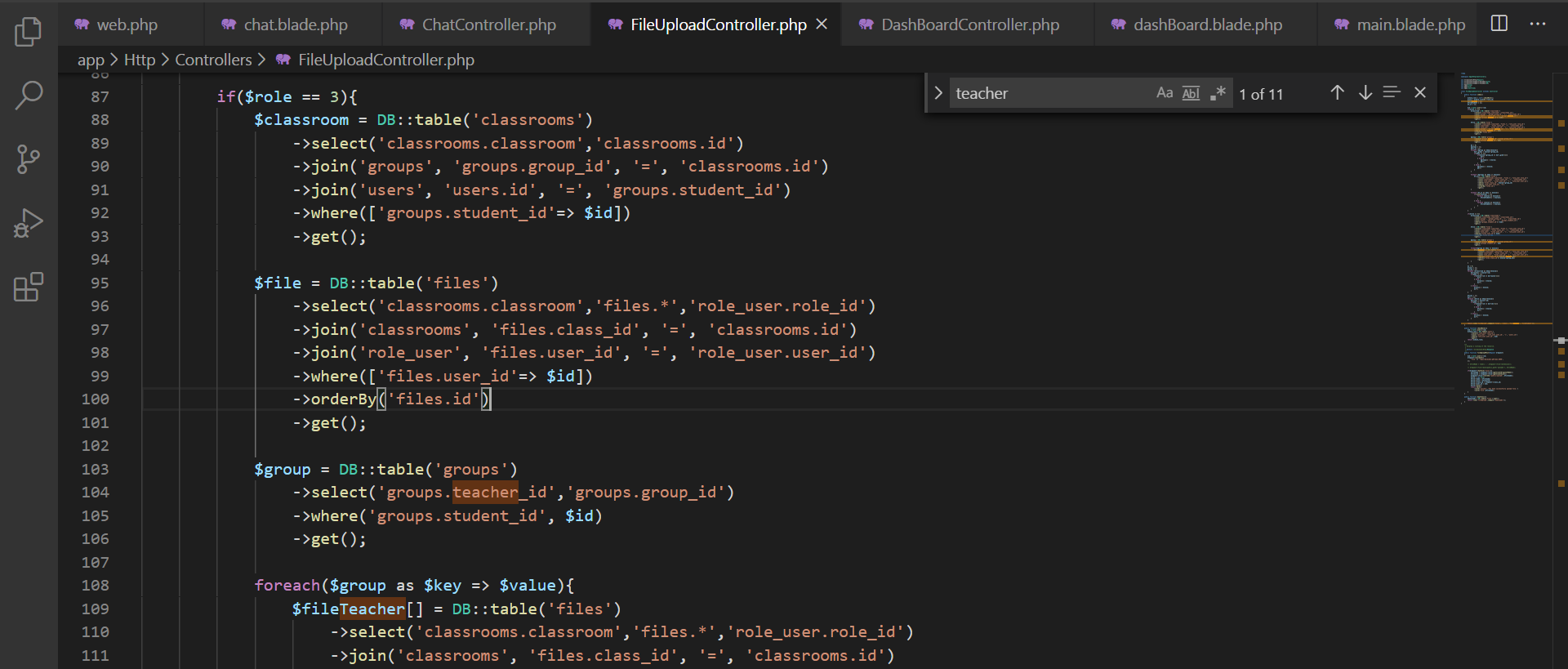


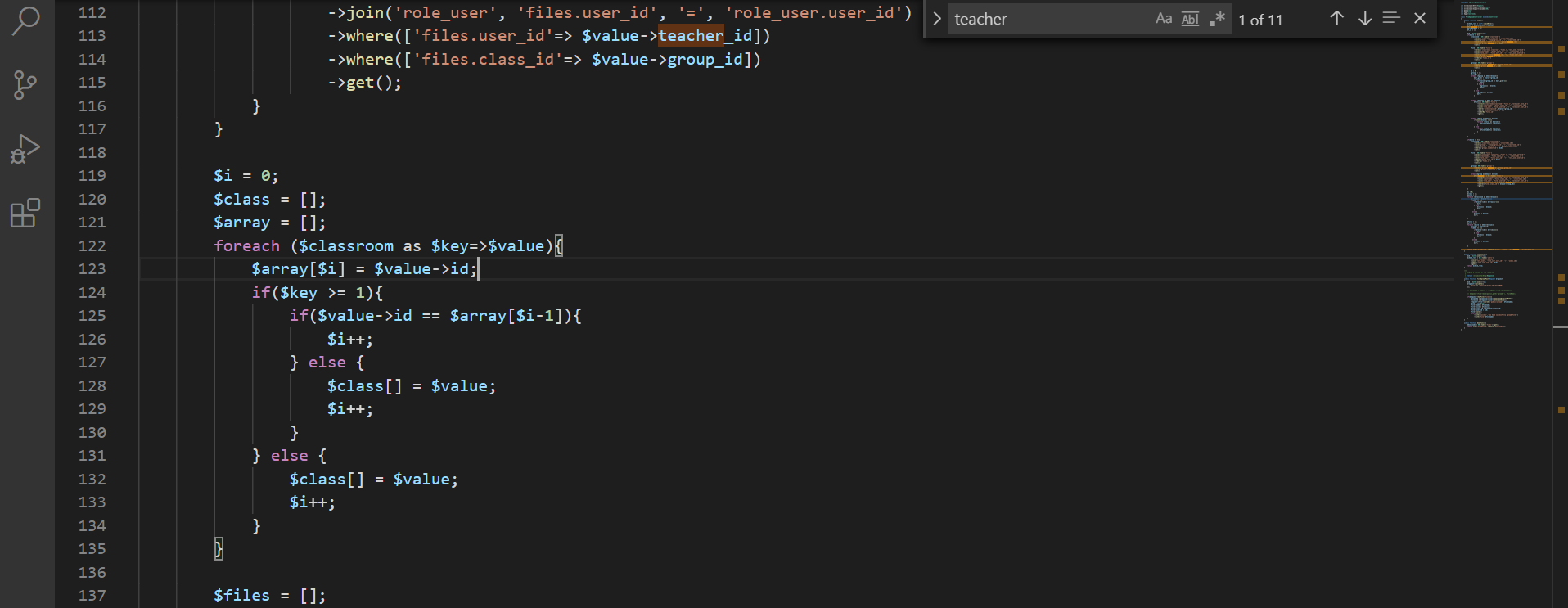


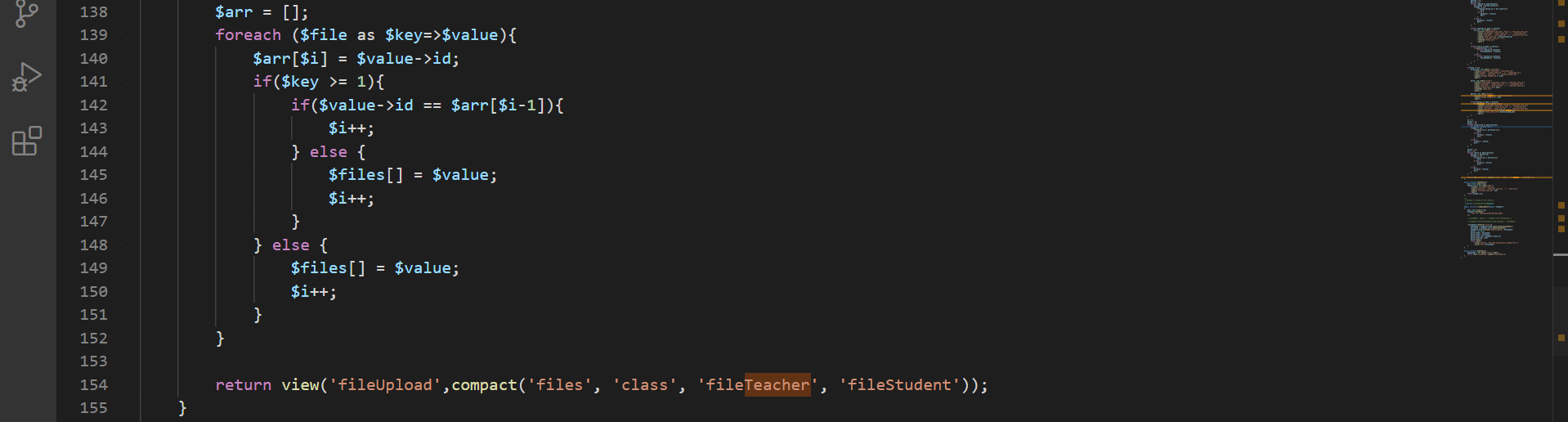




This is code i have show list download with role Teacher. This List Files have show of teacher files and all student files in teacher groups.



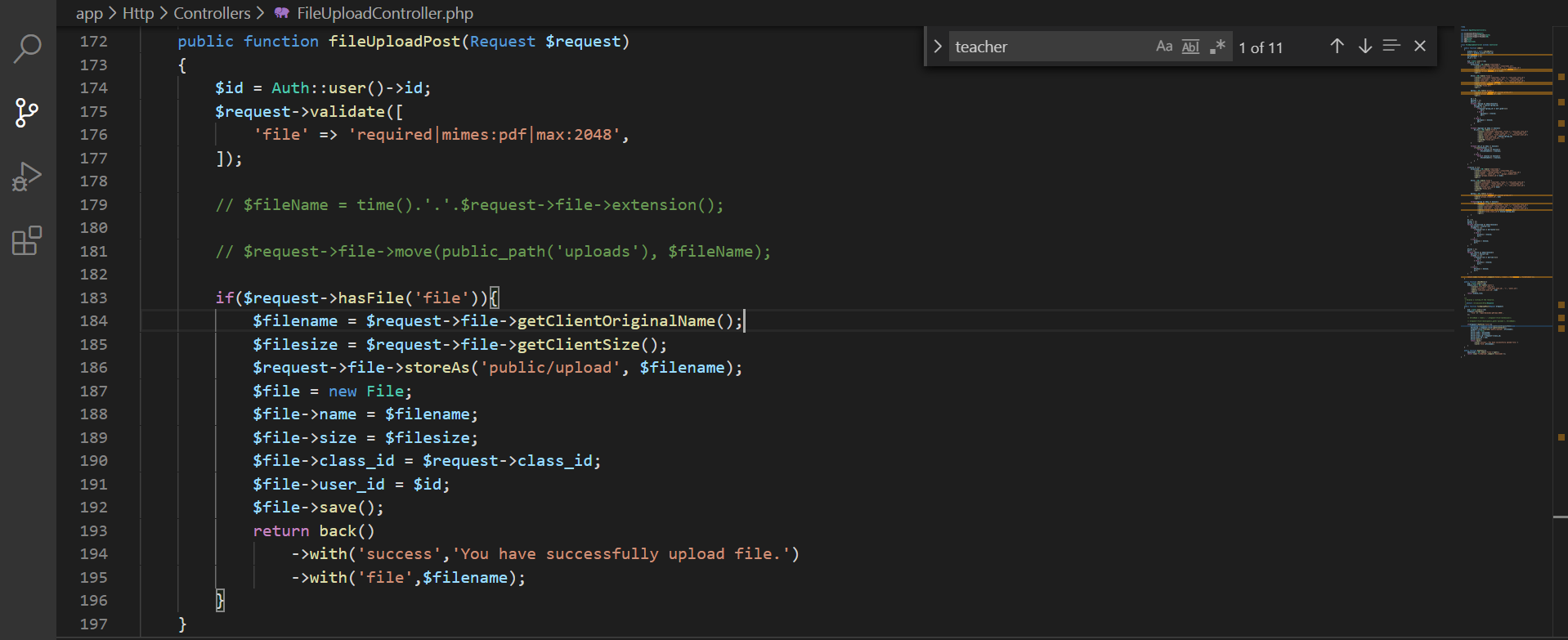


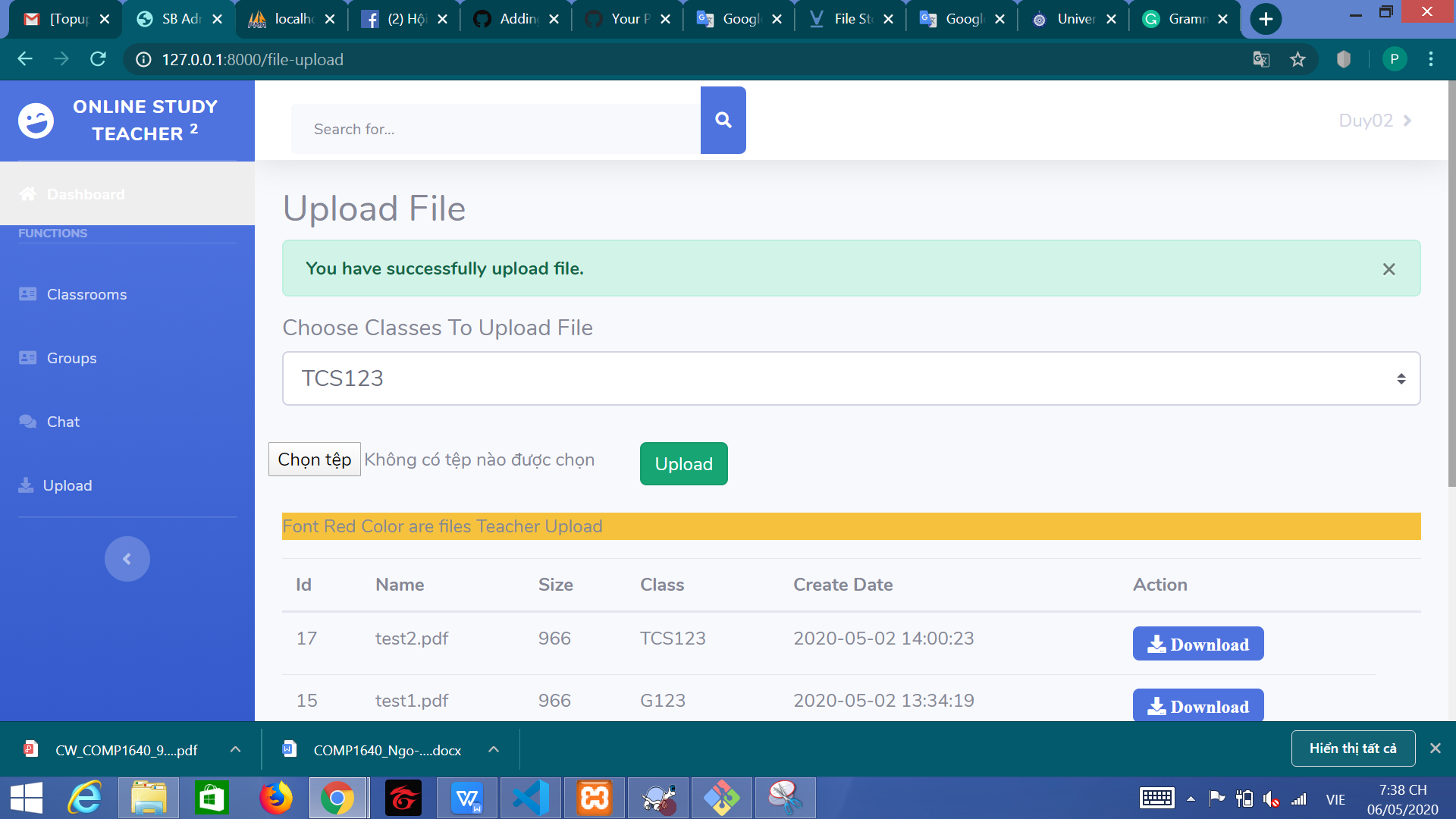


This is code i have show list with role student. In this code i have show files uploads of user when you login and files of teacher in your group you have join.



This is function check role user login.

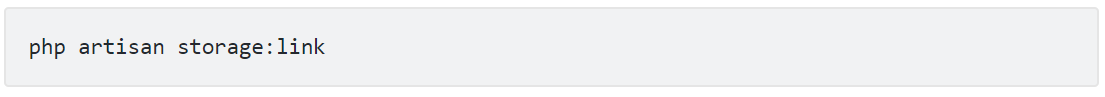




This function have upload files. This function i have check require file paf. I save the url and size of file

If upload file success, i will send message success

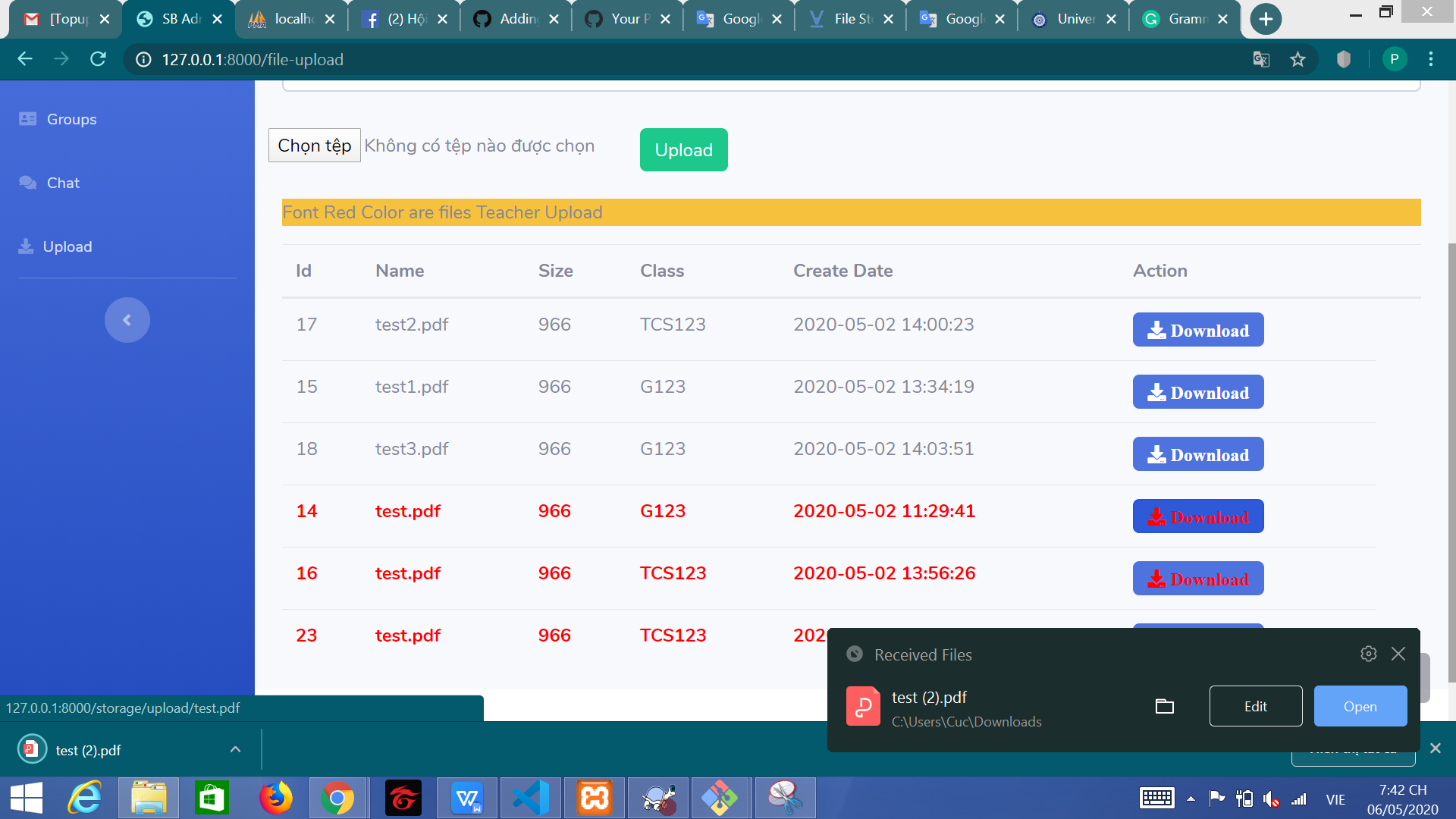
Ans special things in my source code. You need to run Artisan command when you want to download files:



This is the command that must be executed if you want to download files.







This is function have download file. I using Laravel Storage in this stages. If you want to download files, you need to run Artisan command:

# **4. Testing**

## 4.1. Test case

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Description | Expected Outcome | Actual Outcome |
| Login | User input Valid Email and Password | Login successfully and move to dashbroad. | Login successfully and move to dashbroad. |
| User input invalid Email and Password | Login fail and back to login form with error message. | Login fail and back to login form with error message. |
| Staff input Valid Email and Password | Login successfully and move to Staff dashbroad. | Login successfully and move to Staff dashbroad. |
| Teacher input Valid Email and Password | Login successfully and move to Teacher dashbroad. | Login successfully and move to Teacher dashbroad. |
| Student input Valid Email and Password | Login successfully and move to Student dashbroad. | Login successfully and move to Student dashbroad. |
| Menu | After Login, the menu will show based on the role that admin set for the user | - Show everything for role Admin  - Show dashboard, Classroom, Group, Chat for Staff  - Show dashboard, Classroom, Group, Chat, Upload File for Teacher  - Show dashboard, Chat, Upload File for Student | - Show everything for role Admin  - Show dashboard, Classroom, Group, Chat for Staff  - Show dashboard, Classroom, Group, Chat, Upload File for Teacher  - Show dashboard, Chat, Upload File for Student |
| View list user | Admin click into the User management section | Show list of user | Show list of user |
| Add new user | Admin input Valid Email and Password | Add new successfully and move to list user. | Add new successfully and move to list user. |
| User input invalid Email and Password | Add new fail and back to login form with error message. | Add new fail and back to login form with error message. |
| Edit User | Admin edit new role for user | Edit success and return to list user with the updated user role | Edit success and return to list user with the updated user role |
| Delete user | Admin choose user and click Delete | Appear popup to confirm user, user click Yes Delete then delete the user return to list without the deleted user | Appear popup to confirm user, user click Yes Delete the return to list without the deleted user |
| Admin choose user and click Delete | Appear popup to confirm user, user click Close then return to list with full of user | Appear popup to confirm user, user click Close then return to list with full of user |
| View list Classroom | User click into the Classroom section | Show list of Classroom | Show list of Classroom |
| Add new Classroom | User input Classroom name | Add new successfully and move to list Classroom. | Add new successfully and move to list Classroom. |
| Delete Classroom | User choose Classroom and click Delete | Appear popup to confirm user, user click Yes then delete the classroom and return to list without the deleted classroom | Appear popup to confirm user, user click Yes then delete the classroom and return to list without the deleted classroom |
| User choose Classroom and click Delete | Appear popup to confirm user, user click Close then return to list with full of classroom | Appear popup to confirm user, user click Close then return to list with full of classroom |
| View list Group | User click into the Group section | Show list of Classroom | Show list of Classroom |
| Add new Group | User input Group’s Class, Student and Teacher | Add new successfully and move to list Group. | Add new successfully and move to list Group. |
| Delete Group | User choose Group and click Delete | Appear popup to confirm user, user click Yes then delete the Group and return to list without the deleted Group | Appear popup to confirm user, user click Yes then delete the Group and return to list without the deleted Group |
| User choose Group and click Delete | Appear popup to confirm user, user click Close then return to list with full of Group | Appear popup to confirm user, user click Close then return to list with full of Group |
| Chat | Teacher and Student click to chat section | Show the group of class they participate in and the number of unread message the teacher and student that class have | Show the group of class they participate in and the number of unread message the teacher and student that class have |
| Upload File | Teacher and Student click to chat Upload File section | Show the class they participate in and the teacher uploaded file and their uploaded file but not the other student uploaded file | Show the class they participate in and the teacher uploaded file and their uploaded file but not the other student uploaded file |
|  | Teacher and Student choose the file to upload | Done choosing file and click upload, the uploaded file appear to the list | Done choosing file and click upload, the uploaded file appear to the list |

## 4.2. Test result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Description | Expected Outcome | Actual Outcome | Result |
| Login | User input Valid Email and Password | Login successfully and move to dashbroad. | Login successfully and move to dashbroad. | **Pass** |
| User input invalid Email and Password | Login fail and back to login form with error message. | Login fail and back to login form with error message. | **Pass** |
| Staff input Valid Email and Password | Login successfully and move to Staff dashbroad. | Login successfully and move to Staff dashbroad. | **Pass** |
| Teacher input Valid Email and Password | Login successfully and move to Teacher dashbroad. | Login successfully and move to Teacher dashbroad. | **Pass** |
| Student input Valid Email and Password | Login successfully and move to Student dashbroad. | Login successfully and move to Student dashbroad. | **Pass** |
| Menu | After Login, the menu will show based on the role that admin set for the user | - Show everything for role Admin  - Show dashboard, Classroom, Group, Chat for Staff  - Show dashboard, Classroom, Group, Chat, Upload File for Teacher  - Show dashboard, Chat, Upload File for Student | - Show everything for role Admin  - Show dashboard, Classroom, Group, Chat for Staff  - Show dashboard, Classroom, Group, Chat, Upload File for Teacher  - Show dashboard, Chat, Upload File for Student | **Pass** |
| View list user | Admin click into the User management section | Show list of user | Show list of user | **Pass** |
| Add new user | Admin input Valid Email and Password | Add new successfully and move to list user. | Add new successfully and move to list user. | **Pass** |
| User input invalid Email and Password | Add new fail and back to login form with error message. | Add new fail and back to login form with error message. | **Pass** |
| Edit User | Admin edit new role for user | Edit success and return to list user with the updated user role | Edit success and return to list user with the updated user role | **Pass** |
| Delete user | Admin choose user and click Delete | Appear popup to confirm user, user click Yes Delete then delete the user return to list without the deleted user | Appear popup to confirm user, user click Yes Delete the return to list without the deleted user | **Pass** |
| Admin choose user and click Delete | Appear popup to confirm user, user click Close then return to list with full of user | Appear popup to confirm user, user click Close then return to list with full of user | **Pass** |
| View list Classroom | User click into the Classroom section | Show list of Classroom | Show list of Classroom | **Pass** |
| Add new Classroom | User input Classroom name | Add new successfully and move to list Classroom. | Add new successfully and move to list Classroom. | **Pass** |
| Delete Classroom | User choose Classroom and click Delete | Appear popup to confirm user, user click Yes then delete the classroom and return to list without the deleted classroom | Appear popup to confirm user, user click Yes then delete the classroom and return to list without the deleted classroom | **Pass** |
| User choose Classroom and click Delete | Appear popup to confirm user, user click Close then return to list with full of classroom | Appear popup to confirm user, user click Close then return to list with full of classroom | **Pass** |
| View list Group | User click into the Group section | Show list of Classroom | Show list of Classroom | **Pass** |
| Add new Group | User input Group’s Class, Student and Teacher | Add new successfully and move to list Group. | Add new successfully and move to list Group. | **Pass** |
| Delete Group | User choose Group and click Delete | Appear popup to confirm user, user click Yes then delete the Group and return to list without the deleted Group | Appear popup to confirm user, user click Yes then delete the Group and return to list without the deleted Group | **Pass** |
| User choose Group and click Delete | Appear popup to confirm user, user click Close then return to list with full of Group | Appear popup to confirm user, user click Close then return to list with full of Group | **Pass** |
| Chat | Teacher and Student click to chat section | Show the group of class they participate in and the number of unread message the teacher and student that class have | Show the group of class they participate in and the number of unread message the teacher and student that class have | **Pass** |
| Upload File | Teacher and Student click to chat Upload File section | Show the class they participate in and the teacher uploaded file and their uploaded file but not the other student uploaded file | Show the class they participate in and the teacher uploaded file and their uploaded file but not the other student uploaded file | **Pass** |
|  | Teacher and Student choose the file to upload | Done choosing file and click upload, the uploaded file appear to the list | Done choosing file and click upload, the uploaded file appear to the list | **Pass** |

# **5. Agile Method**

**6. Links**

# **TABLE OF FIGURES**

**No table of figures entries found.**

# **TABLE OF REFERENCES**

**There are no sources in the current document.**