Integrated Curriculum of Multi-tier Client/Server Web-Based Database Applications

Ching-Yu Huang

***Abstract*—The Multi-tier Client/Server model is standard and popular for modern computer software systems. For better performance, database servers and web (or application) servers usually run on different machines. The end-user can only access data through the application server, and cannot access the database directly. These servers are often distributed at different locations connected through the network. Web-based architecture is the most essential design for Internet applications. It is very important to teach students the browser-based client/server model in college. Several courses, such as Web Programming, Networking, Operating Systems and Database, cover the related topics, but the curriculum mostly focuses on the individual components. Students seldom have a chance to connect the knowledge from these topics together in a single course. In this paper, an integrated hands-on curriculum is proposed for students to learn the related topics in a systematic method that links these subjects together. Several project topics are also suggested. It is important not only to know these subjects but also to have better learning outcomes by working on projects that integrate these subjects.**

***Index Terms*—Client/server, curriculum, multi-tier, web-based.**

I. Introduction

Distributing service functions on different machines from the user computers is the basic concept of client/server architecture. The client/server architecture is based on the process, program, or software, not on the hardware machine. Beginners are often confused and mix program processes with hardware machines. Clients and servers communicate with each other through the same protocol that could either be on the same machine or on different machines through network.

The key concept of client/server architecture is that the client makes the request and initiates the connection. The server responds and provides the service. The client and server could either be always connected or connectionless, which does not require a session connection between sender and receiver. If the client and server are on two different machines, it is called 2-tier architecture. If 3 machines are involved in the client/server architecture, it is a 3-tier system and one of the machine might be the client and server at the same time. If there are *N* machines involved, it is called an *N*-tier client/server.

The World Wide Web (WWW) can connect a web server (or application server) with many clients (browsers) using Hypertext Transfer Protocol (HTTP) through the network. The e-commerce business is based on the WWW and people use browsers on their laptops to access different kinds of servers every day. For example, we can use a browser to read email, whose contents are stored on Google's Gmail server. An online store such as Amazon allows people to use the browser to place orders where the product, vendor and ordering information are kept on Amazon's servers.

Although the servers can be run on the same machine, industrials usually try to have dedicated machines for providing the important services. For example, the web server and Database Management System (DBMS) are usually not running on the same machine. This is to avoid different servers competing for the same resources (CPU, memory and hard drive) and affect the processing and responding performance. Since the DBMS manages the centralized data, many organizations put the database server inside their firewalls for security reasons and restrict the access to their DBMS only through dedicated applications that can be accessed by the public people.

In the 3-tier client/server architecture as shown in Fig. 1, the PHP programs running the web server could be a client and a server at the same time. It plays as a server to take the requests from the browsers, process and responds with information to the browsers. It also plays as a client and sends the query requests to the DBMS and receives the results back.

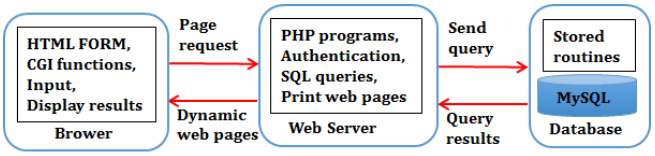

Fig. 1. 3-tier client/server architecture.

The client/server model is very important to e-commerce, which has a strong impact on people's daily life and the market will keep growing. It is predicted that the amount of online retail sales will reach 638 Billion in 2022 [1]. Many applications can automatically send emails with customized information to users and the user can click the links to directly respond back to the applications with individual information. More and more businesses focus on web-based applications that can interact with users by analyzing their personalized data through the multi-tier client/server systems. This paper will focus on how to teach web-based client/server curriculum so students can quickly enter these fields.

II. Student Learning Outcomes (SLO)

The goals of this proposed curriculum are for students to

Manuscript received December 12, 2018; revised March 12, 2019.
The author is with School of Computer Science, Kean University, Union, New Jersey, USA (e-mail: chuang@kean.edu).

doi: 10.18178/ijiet.2019.9.5.1220

learn the fundamental client/server concepts using a web-based 3 tier architecture, do hands-on exercises on the web server and database server, and design a web-based application that will take inputs from the browser, pass the data from the browser to the programs running on the web server and process the data, send SQL query to the database and receive the results returned from database. Since this is a hands-on course, in addition to these goals, students should learn how to design web pages, write programs to handle the browser request, send queries and receive the results. It will be very helpful to teach students how to configure the web server and database server.

After the completion of this curriculum, students should be able to do the followings:

1) Demonstrate an understanding of the web-based database applications and the 3-tier client-server systems and architecture
2) Design and develop interactive client-side web pages that can take user input in different formats and validate the inputs.
3) Design and develop server-side applications run on the web server to communicate with the web pages, and interact with the web server.
4) Design and develop server-side applications run on the web server to access data from database, process the returned results from the database.
5) Integrate the 3-tier web database applications that have interactive front-end web pages using Ajax to work with the back-end programs that can process the data received from the web pages, communicate with the database with input data and returned results, interact with the Operating System on the web server, display updatable results on the browser and save the updated results back to the database.

III. Course Preparation

It is not realistic asking each student to own several different computers to run several different servers. Therefore, the free open source XAMPP [2], [3] software package is adopted in this paper for the demo and exercises. The main user interface is shown in Fig. 2. It stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). The package includes an Apache web server and MariaDB [4], a community-developed fork of the MySQL relational DBMS. The package can be installed on Apple OS-X, MS-Windows, and Linux systems and it doesn't require powerful hardware. In this paper, the browser (1st tier), Apache web server (2nd tier) and MariaDB database server (3rd tier) will be used to simulate the 3-tier architecture, even though they are all running on the same laptop.

The configurations of Apache and MySQL servers can be changed by clicking on the "Config" buttons. /xampp/htdocs is the default document root – a location to keep web pages and files for the Apache web server can access. The default MySQL database files is at /xampp/mysql.

Since the DBMS is the 3rd tier in the proposed curriculum, the basic Structure Query Language (SQL) – SELECT statement will be covered. In order for a computer language to communicate with the database, the language should provide the database functions to connect to the DBMS, send SQL queries, and receive the results back from the DBMS. PHP has built-in MySQL functions [5] that can connect to MariaDB and interact with the database. Students won't need to install any other libraries for working on client/server projects. The Apache web server and MySQL database server can be easily started and stopped by clicking on the corresponding buttons shown in Fig. 2.

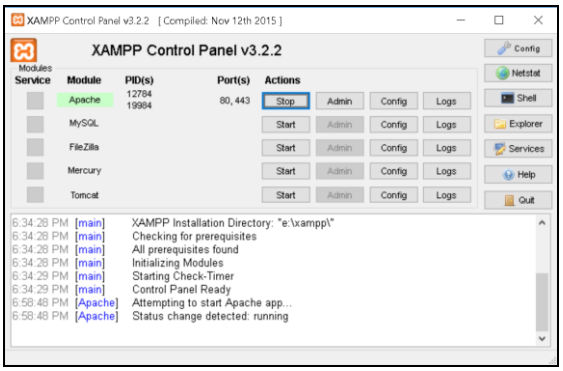

Fig. 2. The main graphic user interface of XAMPP.

The prerequisite for students to study this curriculum is to have programming skills. Students should know the basic variables, data types, assignments, variable scope, logic comparison, if-else, loops, 1-D and 2-D arrays, and methods (functions). PHP and Perl are server-side scripting languages and the package provides a terminal for running the scripting programs as shown in Fig. 3. In this paper, PHP will be the computer language for the demo and exercises.

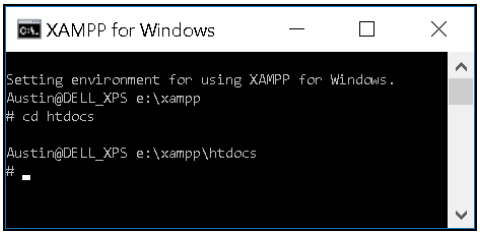

Fig. 3. The shell terminal of XAMPP.

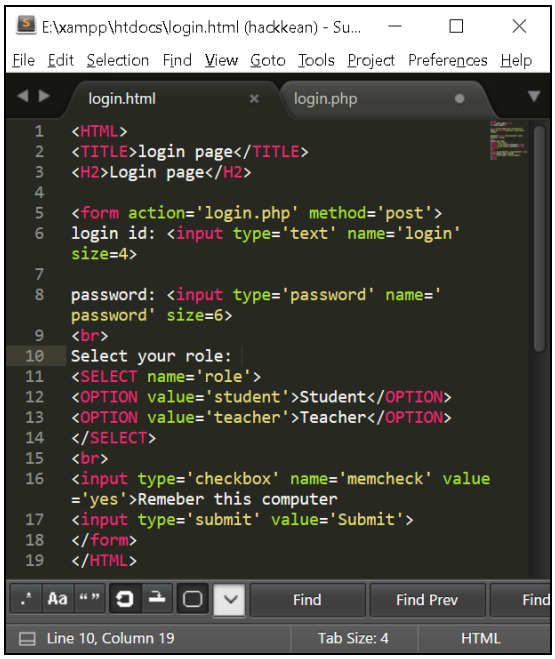

Fig. 4. The Sublime text editor shows the client source code running on the browser - a HTML program login.html.

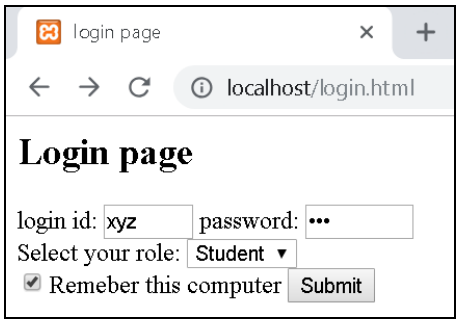

Fig. 5. The web page of the HTML code in Fig. 4.

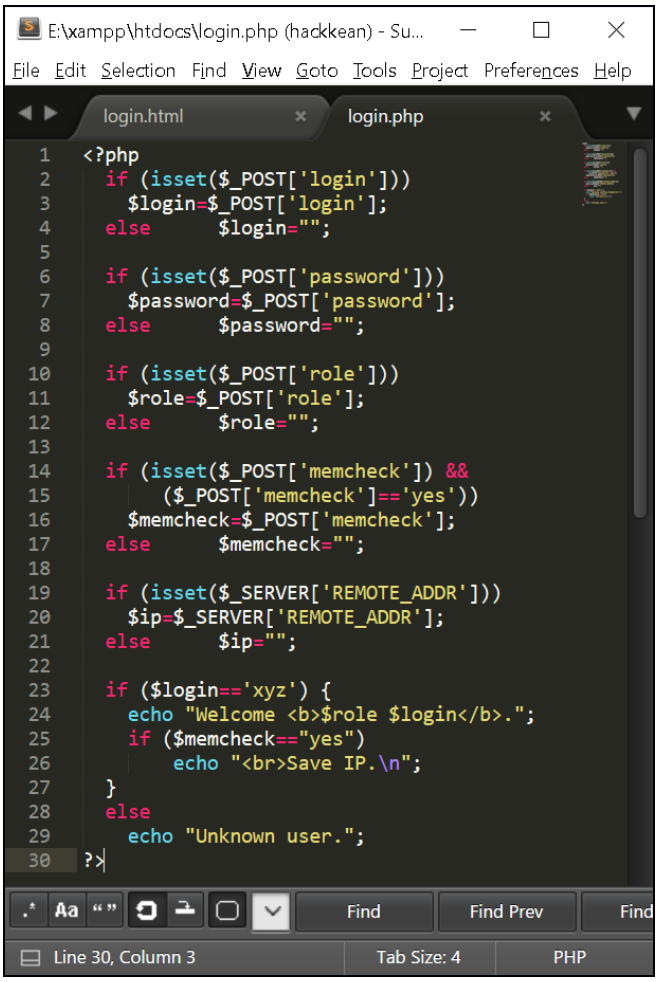

Fig. 6. The Sublime text editor shows the source code running on the web server - a PHP program login.php at the 2nd tab.

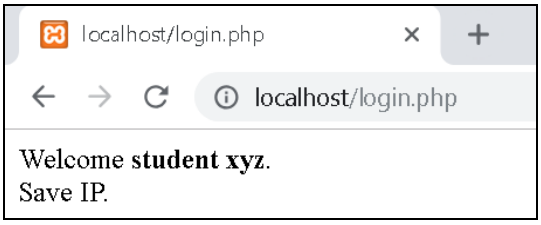

Fig. 7. The result shows on the browser is generated by the PHP program after the Submit button is clicked.

A good editor is very helpful for the programming and debugging. Sublime Text [6] is a free proprietary cross-platform source code editor supporting many programming languages included the HTML. It shows the line numbers, keywords and user defined variables in different colors. HTML source code for a simple login web page is shown in Fig. 4, and the corresponding web page is shown in Fig. 5. The login.php program shown in Fig. 6 will be called when the "Submit" is clicked and the PHP program will display the result on the browser as shown in Fig. 7. In this paper, all the example source codes are displayed in Sublime Text.

IV. Curriculum

The proposed curriculum is based on 16 weeks of classes with 2.5 hours per week. The total number of hours is 40, including 35 hours of lecture and 5 hours of exams and reviewing answers. Students should also spend 80-120 hours outside the classes on studying and completing homework and project assignments. There are 5 major components that should be covered – fundamentals of client/server, client-side web design and programming, server-side application programming, server-side database programming and integration between these components. These topics should be taught in sequence.

A. Fundamentals of Client/Server

The definition of client and server should be covered. This component should have 2 lecture classes, totaling around 5 hours. Several real-world examples should be given to demonstrate client/server concepts such as Google shared drive, product searching in Amazon, Weather Channel, etc. The following sub-topics should be discussed when demonstrating the examples:

- Client/server, process, protocol, daemon, and 3-tier architecture concepts
- How to identify who is the client or server
- Install XAMPP, Sublime Text, Chrome browser
- Identify the input/output and resources (memory, CPU and storage) used by the client/server system.
- What functions/services does the server do/provide?
- What is dynamic web page? Who provides the updated content on the web pages?
- Internet Protocol (IP) address, and how to show the IP address of students' own laptops using "ifconfig -a" command on Linux and Mac terminal or "ipconfig/all" command on MS-Windows command prompt window (cmd).
- Examine the processes running on the systems and meaning of each information – user id, process id, running time, command, etc. "tasklist" command on MS-Windows and "ps -ef" command on Mac and Linux systems.
- Exam and set the file permission mode – owner, group, public and the read, write, execute mode.

B. Client-Side Web Design and Programming

The client-side programming can be easily demonstrated by using HTML to design web pages. This component should have 3 classes, totaling 7.5 hours of lecture. Students should do hands-on HTML and JavaScript [7], [8] coding, such as making the login web page as shown in Fig. 4. The following topics should be covered:

- Concepts and structure of HTML Document Object Model (DOM).
- Basic HTML tags and related attributes – <FONT>, <A HREF>, <SELECT>, <INPUT>, <BR>, <TITLE>,

<IMG>, <UL>, <TABLE>, <TR>, <TD>, <DIV>, <FORM>

- HTML <INPUT> tag and attributes in array style.
- Common Gateway Interface (CGI) – POST, GET methods.
- Define the server-side PHP program in HTML <FORM> and the PHP program location.
- Basic JavaScript – variables, array, function, mouse events and handlings, getElementById, time and utility functions
- JavaScript debugging using browser tools (e.g. Chrome console)
- Data validation (type, empty and range) on the client-side (front-end)
- Utilizing 3rd party JavaScript libraries or APIs, such as Google Maps and Google Charts.

Students should be able to design web pages that have input boxes for login and password, check box, dropdown list, etc different input formats as shown in Fig. 5.

C. Server-Side Application Programming

This component works on the 1st tier client browser and the 2nd tier web server. The server-side programming requires students to have better coding skills than on the client-side because it usually uses a host scripting language such as Python, Perl or PHP. That is why students should be good at programming prior taking the course. It is very important to let students know the difference between client-side and server-side programming, and which programs and machines handle the events and the corresponding processes. This component should have 3 classes with 7.5 hours of lecture. The following topics should be covered:

- Receiving data from the browser using $_POST[] and $_GET[] methods.
- Using PHP to receive multiple array values from the <INPUT> array style.
- Receiving client information from the web server using $_SERVER[], such as IP address, browser and OS type.
- Utilizing the client information to protect and improve the security for the web server.
- Retrieving the date and time from the browser and the web server. Set and convert the time zones.
- Permission mode of the PHP programs and folders.
- The default path and the system environment when the programs run on the server.
- Opening files to read/write the files.
- Changing the permission mode and ownership of the output files.
- Uploading files from HTML, identifying the file type, and saving and moving the file into a specific folder on the server.
- Running other programs on the server using exec_shell() function.
- Passing messages or database results from the PHP programs to the JavaScript on the HTML web pages.
- Identifying the causes of system error messages shown on the browser when the PHP program is run the web.
- Debugging the PHP programs on the command line as shown in Fig. 8.
- The basic concepts of Operating System (OS), web server, and their relationship.
- Utilize the PHP sendmail() function to email users from the PHP programs.

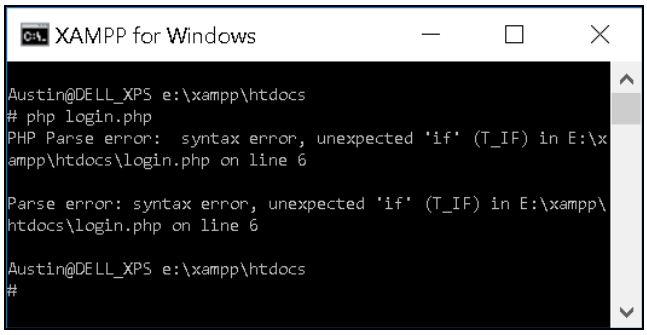

Fig. 8. It shows how to check the syntax of a PHP program and execute the program using the Shell command line. A syntax error was caught and the message is displayed on the command line.

D. Server-Side Database Programming

This component is the extension of server-side programming by adding additional database components. However, the PHP programs on the 2nd tier web server is considered as a client to the 3rd tier database server. Students should be able to send the SQL query request from the PHP programs to MySQL database, receive and process the return results. This component should have 3 classes with 7.5 hours of lecture and cover the following topics:

- *Static and dynamic web pages.*
- *Basic SQL DML – SELECT, INSERT, UPDATE, DELTE and WHERE clauses.*
- Basic *SQL joins with multiple tables.*
- Basic *SQL aggregate functions – count(), sum(), avg(), max(), and min().*
- *PHP MySQL functions – mysqi_connect(), mysqli_close(),* mysqli_*query(), mysqli_select_db(), mysqli_fetch_array(),mysqli_num_rows()mysqli_free_result(), mysqli_affected_rows(), and other related functions.*
- Displaying *the database results in HTML <TABLE>, <SELECT>, <INPUT> tags on the browser.*
- Identifying *the no results caused by empty result returned from the database or SQL query error.*
- Debugging *the SQL query error – syntax, logic or others.*
- Error *handling and error message.*
- Password *encryption and comparison.*
- Getting *the new auto-increment id number using mysqli_insert_id() when INSERT a new item.*
- Since *the SQL INSERT, DELETE and UPDATE will return no results, how to identify each if they are successfully executed.*

E. Integration

The definition of client and server should be covered. This component should have 3 classes with 7.5 hours of lecture.

- Authentication and authorization by comparing the login, password received from the browser against the login and encrypted password in the database.
- Asynchronous web applications using Ajax with callback functions.
- Ajax jQuery

- Passing data from the browser to PHP using Ajax POST or GET methods, and included the data as variables in the SQL query.
- Using 3rd party JavaScript library/package to display the processed database results on the browser.
- Using database results to update a specific browser content without reloading the whole web page.
- Setting and converting the time zone among the 3-tier systems – the customer delivery address, vendors, and the system web and database servers.
- Displaying the results from the database on the browser for users to select or change the content and save the changes back to the original records in the database.
- Understanding the data flow and client/server relationship among the 1st tier, 2nd tier and 3rd tier.
- Identifying and utilizing the different time among the 3-tier programs.

V. Methods of Assessments

It is important to assess students' studying, identify their weak spots and adjust the teaching pace. The assessments include 3 parts – written exams, homework, and projects. The exams should be around 60% of total final points. Homework and project assignments should be 20% each.

A. Exams

There should be two quizzes (10 points/each), one midterm (25 points) and one final assignment (25 points) with two phases, or two different smaller projects. Students are expected to spend total 40 - 60 hours on studying. It is very import that instructors review answers with students. The exams should include the conceptual questions, hands-on skills, analysis, design and development of the projects.

B. Projects

There should be one project assignment with three phases, or three different smaller projects. The projects should involve with 3-tier design and development – an interactive user interface and display on the front-end browser as the 1st-tier, 2nd-tier programs run on the web server to receive data from the browsers, process the data, interact with the server, save data into the 3rd-tier database, receive the data from the database and display the processed results on the browser. The total points for all the projects will be 30 points. Students are expected to spend total 40 - 60 hours on the projects. Here are some examples:

- Design and develop a mini-online store allows customers, vendors to sign up, login, add products, place orders on multiple products, and view the order history and sales report. Students need to convert the time zones that could be different among these entities.
- Design and develop a web password reset system that users can sign up, login, and securely reset the encrypted password on the database through the web browser. The system should send email to users to reset the password and confirm the password is changed. The activities should be logged.
- Design and develop a web system that allows users to sign up, login, enter the travel information (trip name, date, multiple cities, states, countries, notes) on the browser and save the data into the database. Then, users can select the trip name to display the trip information and locations on the Google Maps.
- Design and develop a web cybersecurity data visualization system that allow users to upload secure or Apache log files, process the files to extract the IP address, date and time, and other data, and save the information into the database. Then, users can select and visualize the data on the browsers in Google Charts – Table, Line Chart, and Pie Chart formats, and view the data on the Google Maps.
- Design and develop a web image management system that allow users to upload images, save and move the image to specific folders, shrink the images into thumbnail, and save the image information (name, path, upload date, owner name) into the database. Then, users can select and view the image on the browser.
- Design and develop a web system that users can copy/paste or upload a Java source code, save the code into a file, compile and run the program. The system should show the error message on the browser if there is any compile or run time error. If there is no error, the system should show the results of running the Java program on the browser.
- Design and develop a web system that can ping and detect whether the given server names are alive or not, and display the status in different colors [9] on the browser.

C. Extra Study

If students wish to do the project on a 2 or 3-tier architecture on 2 different operating systems, they can consider to use the virtual machine. Oracle Virtualbox [10] is a virtual machine that can host different kinds of Operating Systems, such as Ubuntu [11], which is a popular Linux Operating System. A Shell prompt is shown in Fig. 9 with Virtualbox version 5.2.18 and Ubuntu Linux server 18.04.1-live. The Linux system is a better Operating System than MS-Windows for students to learn the server environment in the following topics:

- Process status
- Scheduling – cron job
- Permission mode
- Context switch
- File systems

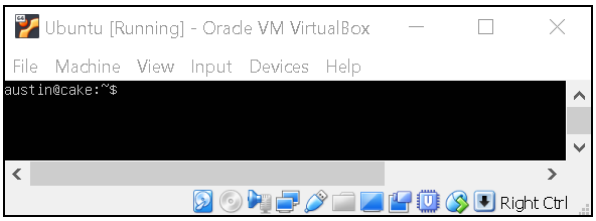

Fig. 9. The Ubuntu Linux system is installed on the Virturalbox. The $ is the Linux command shell prompt.

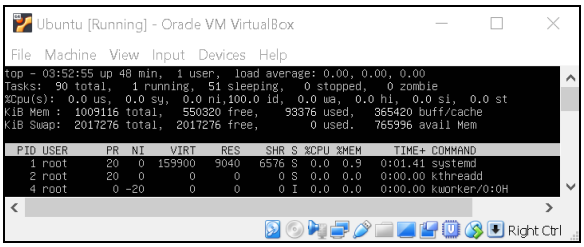

Fig. 10. The Linux "top" command can show the process and system running information.

The top command provides more details about the process statuses, process information, virtual memory, and other system related information as shown in Fig. 10.

In the normal 3-tier architecture, the client and web server should be on different operating systems and PHP programs must be run by the web server. A tool is required to transfer the files from the client to the server. FileZilla as shown in Fig. 11 is convenient to transfer files between two operating systems – the local system is the client and the virtual system is the server.

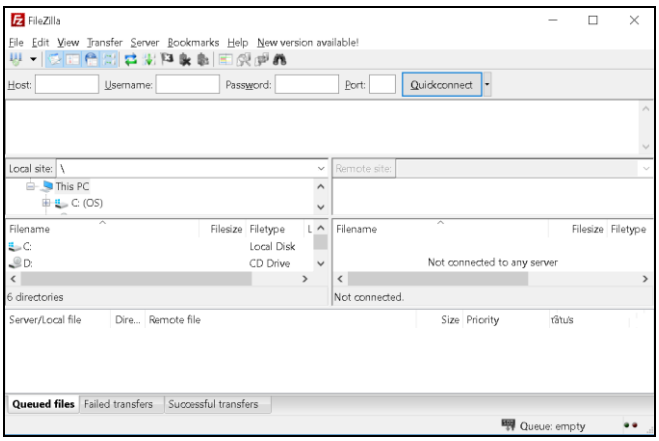

Fig. 11. The FileZilla software allows users to transfer files between two different Operating Systems.

VI. Conclusions

Client/Server architecture is an industrial standard in the modern software design. Web-based database applications provide flexible and scalable features to organizations. This paper defines the 5 SLOs and a multi-tier web-based client/server curriculum was proposed for students to learn 5 main components – concepts, client-side programming, server-side programming, server-side database programming and integration to cover both conceptual knowledge and hands-on skills. In this course, the hands-on projects are very important to evaluate students' studying. 5 project assignments are provided as examples for students to exercise. Students can utilize the XAMPP package and other software used in this paper for working on the projects.

References

[1] Retail e-commerce sales in the United States from 2016 to 2022. (2018). [Online]. Available: https://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/

[2] XAMPP. (2019). [Online]. Available: https://www.apachefriends.org/index.html

[3] XAMPP Wiki. (2019). [Online]. Available: https://en.wikipedia.org/wiki/XAMPP

[4] D. B. Maria. (2018). [Online]. Available: https://mariadb.org/

[5] Introduction to PHP's built-in MYSQL functions. (2018). [Online]. Available: http://www.phpforkids.com/php/php-mysql-database-functions.php

[6] Sublime text. (2018). [Online]. Available: https://www.sublimetext.com/

[7] JavaScript tutorial. (2019). w3schools.com. [Online]. Available: https://www.w3schools.com/js/

[8] Online JavaScript Editor. Tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/online_javascript_editor.php

[9] HTML color. [Online]. Available: http://htmlcolorcodes.com/

[10] Virutalbox, Oracle. (2019). [Online]. Available: https://www.virtualbox.org/

[11] Ubuntu. (2019). [Online]. Available: https://www.ubuntu.com/

Ching-Yu Huang is an assistant professor of the School of Computer Science at Kean University, Union, New Jersey, USA since September 2014. Dr. Huang received a Ph.D. in computer & information science from New Jersey Institute of Technology, Newark, New Jersey, USA.

Prior to joining Kean University, Dr. Huang had more than 16 years of experience in the industry and academics in software development and R&D in bioinformatics. His research focuses SNP genotype calling and cluster detection; image processing and pattern recognition, especially in microarray and fingerprint; geotagged images and location information reconstruction; database application development; data processing automation; e-learning, educational multimedia, methodology, and online tools for secondary schools and colleges. Dr. Huang has more than 40 publications in journals and conferences and more than 20 presentations in workshops and invited lectures.