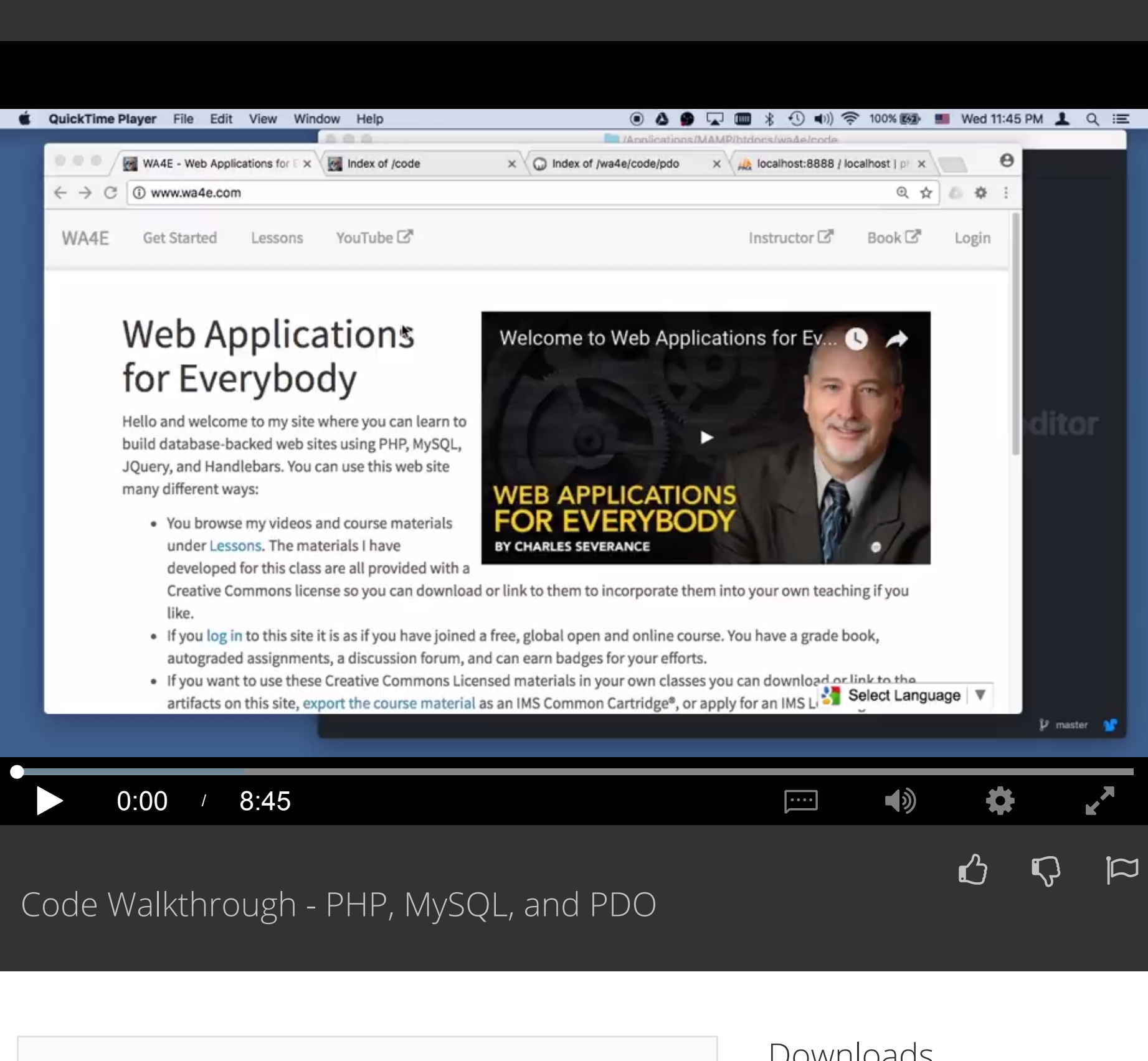


Lecture Content

PHP Database Libraries	7 min
Running SQL Queries in PHP	11 min
Accessing MySQL Using PDO: Inserting Data	10 min
Security Issue: Avoiding SQL Injection	9 min
Error Handling with PDO	8 min
Code Walkthrough - PHP, MySQL, and PDO	8 min
Code Walkthrough - Inserting and Deleting Data	9 min
Code Walkthrough - Security and SQL Injection	8 min
Practice Quiz: PDO	5 questions

Assignments

Bonus Materials



Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

0:00

Hello everybody and welcome to another code walk through on Web Applications for Everybody. The code that we're going to walk through today is the code for PDO. Now one of things that'll be interesting is everything we've done up to now you can actually run on Web Applications for Everybody. Since this one needs a database, and I didn't set the database up for everybody to share on web applications, you actually gotta download the pdo.zip and get it somewhere onto your local hard drive. So, I've got actually all the codes sitting right here, and so I've got all the code here. So you'll down pdo.zip and you'll extract it somewhere in your HT docs in a way that you can find things, and then eventually you'll have this stuff on your local host, so code/pdo that's all those files so that now I can run these things. This AA error check thing checks to see if you got display errors off, so that's nice. If you get a different you might want to click on that and just see if it checks your errors. Always is a good idea. So one of the things that's important is it's got some notes in here, and you gotta run some SQL commands, and so I'm going to make a database here. So I'll do some SQL, go back to these notes, create database misc, yada yada. Make a thing called fred and zap, otherwise the things won't work.

1:18

Now I've got a new database so I go into that misc database. If I am on command line or in Linux I might have to say USE misc, and then I want to create some tables and they're going to have user\_id, name, email, password, and want to make the logical key be email. By now you should know SQL, and so this ought to be completely normal. Boom I don't know what was mad about that. What was it complaining about when I typed that in?

1:46

A comma or closing bracket was expected near key. But it worked. I don't know.

1:55

Clear.

1:57

Now, and the last thing I've got to do from the notes is insert a couple of users, Chuck and Glen into it. Glen is a real person by the way. I'm a real person. So now in database misc, I've got users table, and if I take a look at that, I got two user accounts, so there we go. Now we can start playing with some php.

2:22

So, let's take a look at the very first bit of code. So this is important.

2:31

So I'm going to put out a pretag and then I'm going to call and construct a PDO object using the PDO class, and there are three parameters to construct it. The first parameter's what's called a connection string, and it tells what kind of database we're going to make, be connecting to, which is MySQL. Where it's connected now, most of the time in production you put your database on a different server, and it's got an address, your local host just means we're the same server and we are, and then there's a port. Now, if you're using XAMPP on Windows or Linux you're probably going to choose 3306 here, instead of 808089, and then you have to have the name of the database, and that was the name of the database that I just created, the misk database. Okay, and then you have to have the account and the password, and I did that in these notes. I made the account name Fred, and I named the main password zap, and you can just do this on your local computer, it doesn't really, there's not a lot of security holes of a database server, because you have to be coming from localhost or from your, 127.0.0.1 is another version of localhost, and so you can't connect from the outside world into this database, unless your computer's been completely compromised and then it just doesn't matter. So, you don't really have to be too secure about this. Okay, so, this creates a connection it doesn't move any data. It just is a way for us to send SQL commands back and forth, and then we can use this returned object and then call the query method and send some SQL. So this is the same as SELECT \* FROM users, that we would type.

4:16

[NOISE] There we go. The same SELECT \* FROM users, and say go, and it returns these four rows as a record set, and then what we get back here in php is a statement and then we can use that statement to fetch one row at a time using (PDO::FETCH\_ASSOC) array, says we would like each row given to us as a series of key value pairs where the name of the key is, come back, the name of the key is the name of the column, user\_id equals 1, name equals Chuck, email equals csev, password equals 123, and so, and this will give us a false, row will give us a false, which will stop the Y loop. So, this is a quick way to loop through all of the rows that are going to be retrieved and print them out with print\_r{\$row}. Now, if this database doesn't exist, or something is wrong, this will cause trouble. But, now I've got it all set. Find, I can say goodbye to that, and if I run first, it should show that. Now, there's lots of things I could change. Let me just change something here. If I change that to be the wrong password, we'll see what happens here. So I run first and it blows up, and that's because, and this is why errors are turned off, because I just blurted my password to my database. So I'll fix that and it will work, and so this PDO, this will blow up if you don't have these things right, and so sometimes making this stuff work, when you first start out, that is the hardest line to get working. Once you have that working the SQL usually is correct. Now, it will make a different mistake, if I make a mistake with the SQL it will blow up a different way, it'll say, you got some bad, SQL there. So I'll fix that. We'll talk a little bit more later about how errors work. So there we go. Now another way that we do this is we don't necessarily use print\_r, and this is a good example of reading data from the database, and turning it into a table, and we're just using echo statements cleverly to construct all of this data. So if I go to second, It looks nice, and if you just do a View Source, you just see that there's table, and then there's a beginning of the table, and then the data that came from the database, the end of a table data, a new one, data that came from the database, on and on, and on, and on, and on, and so we just construct, we loop through and instead of just dumping it we actually produce some kind of pretty meaningful HTML.

6:58

Now another thing that we do is we tend not to want in to put this statement in every line of every file that we're ever going to do, because a lot of websites have hundreds of files, and we don't want to put the password, and all the database connection information into every single file. So we refactor it and we use a require statement to pull in another file. The rest of this is just a read to read and print all this stuff. So if I look at third here, Boom, out she comes. But then I look in pdo.php and all this does is sets up this variable \$pdo. I could name this anything, I just happened to name it \$pdo. The class is capital PDO and \$pdo is the variable that I made, and I'm setting a error attribute, and I'll talk about this in a bit. This ERRMODE, I'm saying be aggressive with your errors and blow up and stop, rather than continue on. So, that is this, all we did here between first and third,

8:01

First we did it here, and in third we turned that into, oops, let me select, and in third we just did it all in require and put it in a separate thing, and it's so you don't leak the passwords and that's just a really a general good pattern. So that gets us sort of started talking about how we make a connection to the database, and how we send an SQL command, and how we loop through the data that we get back. [MUSIC]

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT

Transcript (English)	txt
----------------------	-----

Would you like to [help us translate](#) the transcript and subtitles into additional languages?