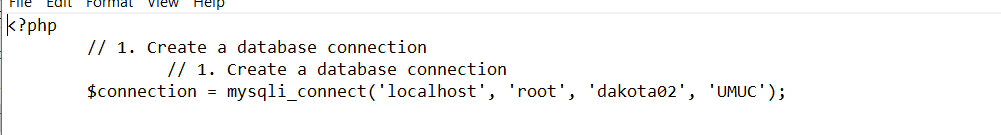
**Use of hard-coded credentials**-

Hard-coding a secret password or cryptographic key into your application is bad practice. It is always a bad idea. If PHP ever fails to process the page, it could be dumped as plain text to the user. The damage to a company can be significant if customers find their accounts were hacked. The bad press alone can cost a company a great deal of money and credibility.

For this demonstration, we will keep a password separate from our code. We will store the password in a separate file and let our code reference that file. We would obviously want to keep that file protected as well. A location that is not accessible from outside of your server is best. If our PHP code is displayed as text, we won’t expose any passwords.

Vulnerable-

We don’t want anyone to be able to see things like our username, password, or even the name of our database. We can see them on full display here.



We should have out information in a separate file. One that cannot be accessed outside of the server.

Mitigated-

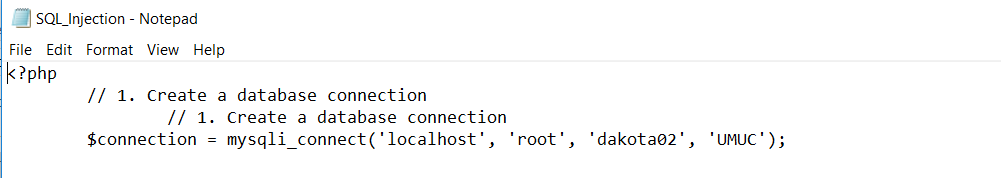
We are going to store a database connect script in a different file, outside of where an end user can get to it from a server. I will use my previous SQL Injection project in order to demonstrate this. It will work well and be easy to show how this problem is solved.

We will follow these simple steps:

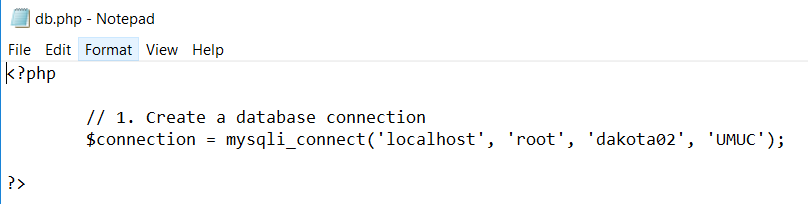
1. Cut the connect script from the SQL\_Injection file.

2. Paste it into a new file by itself called ‘db.php’. This file will be stored outside of the web server.

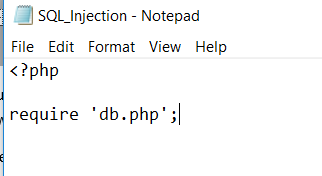
3. In the original SQL\_Injection file, add a bit of code at the top that tells the application to include the ‘db.php’ file that includes our database connection. ( require ‘db.php’; )



Shows original script in SQL\_Injection file.

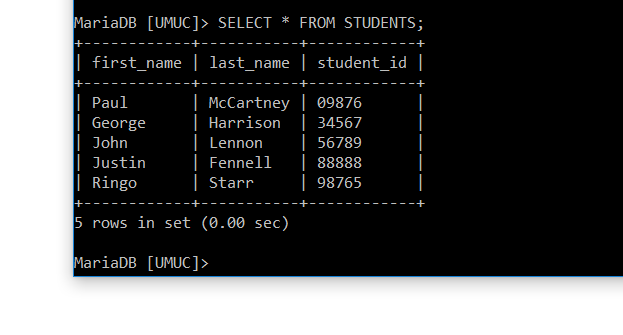


After copy and paste. Shows our new db.php file containing our code to connect to the database.



Shows the line of code we added to the top of the SQL\_Injection file.

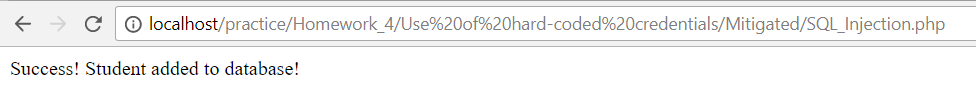
Now to test our new script. We will try to add a name to the students table using the form. If the database fails to connect, we will get an error.



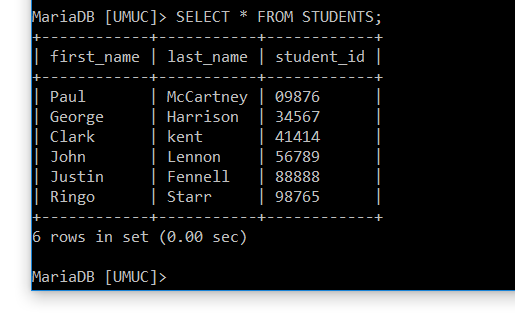
Shows the database



Our form we will attempt to submit. Hopefully we do not get an error.



No errors! Let’s check to be sure it worked.



Success!