CST8285

PHP & MySQL

Overview

- PHP provides core functionality for web programming
- PHP extensions allow for increased capability of PHP programs
- Some examples of extensions include:
  - Calendar – extension for working with dates
  - SimpleXML – converts xml documents to objects
  - MySQLi – Allows PHP to interact with a MySQL database.

MySQLi Extension

- Builds off the original MySQL extension
  - MySQLi stands for MySQL improved
- Provides an API for PHP applications to access and use a MySQL database
- Installed by default as of PHP5

Procedural vs Object Oriented

- MySQLi provides an interface for both procedural and object-oriented programming paradigms.
- The use of either is based on developer preference.
- In this course, we will be using the Object-Oriented paradigm.

Before we start

- From this point on, MySQL will need to be functioning properly on your PC.
- If you are having issues with MySQL, **please do not wait until the day your lab is due to tell me.** No lab extensions will be given due to issues with MySQL.

The MySQLi Object

- Contains methods to connect to, query, and administer a database.
- Constructor takes minimum 4 arguments
  - Database host (eg: 127.0.0.1)
  - Username – MySQL username
  - Password – MySQL password
  - Database Name
- Ex: `$mysqli = new mysqli('127.0.0.1', 'root', 'password', 'db_name');`
  - The above code will create a mysqli object and establish a database connection.

Important MySQLi Properties

- The MySQLi object provides properties with details about connections, queries, results, and more.
- Some important properties to note:
  - connect_errno and connect_error
    - Provides information about connection errors
  - errno and error
    - Provides information about non-connection related errors
  - affected_rows
    - Provides the number of rows affected by the previous query.

Important MySQLi Methods

- query()
  - Accepts an SQL query and returns a mysqli_result object
- prepare()
  - Accepts an SQL query returns a statement object
- For more information about the mysqli class, please visit http://php.net/manual/en/class.mysqli.php

The mysqli_stmt class

- Contains methods to prepare and execute a prepared statement against a MySQL database.
- Properties
  - error & errno – contains information about errors encountered during execution
  - affected_rows – contains the number of rows affected by the previous INSERT, UPDATE, or DELETE statement.

The mysqli_stmt class

- Methods
  - prepare()
    - prepares an SQL statement for execution
  - bind_param()
    - used to bind variables to a prepared statement as parameters
  - execute()
    - executes a prepared statement
  - get_result()
    - returns a result set from the previous query execution

The mysqli_result class

- Holds the result of a database query
- Key properties
  - num_rows – indicates the number of rows in the result
  - field_count – indicates the number of fields in the result (columns)

The mysqli_result class

- Key methods:
  - fetch_assoc()
    - returns the next row in the result as an associative array (column_name=>value)
  - free()
    - Clears the memory holding the result
  - fetch_row()
    - Returns the next row as an enumerated array (ie: integer key starting at zero)

Closing Connections

- Although the MySQLi class provides a method to manually close database connections, it is not necessary to use. This is because **MySQLi will automatically close the database connection at the end of the script.**

Sample Application

- In BlackBoard, there is a sample application that can be found under Course Documents -> Week 11 -> Week 11 Lecture. Download the file called Week11Demo.zip.
- Extract the Week11Demo folder to your htdocs folder (or whichever directory is used by your server)

Setting up the database

- In the Week11Demo project, there is a folder called **sql**. Inside that folder is a sql script called **initCST8285.sql**
- Using phpMyAdmin (or the MySQL workbench, or the command line), run that script to create the CST8285 database and database user.

Running the application

- If your project is set up correctly, you should see the following when you navigate to http://localhost/week11demo/index.php: