Lab 08 – Book O Rama PDO remix

Getting Started

Visual Studio Code

1. Download the lab template from Blackboard
2. Extract it
3. Create a database called **Lab08**
4. Execute the books.sql file to create the tables
5. Execute the book_insert.sql to populate the tables above

Solution

Based on the Demo Code and the bookorama database. Create a web application that allows you to add and delete Customers to/from the database.

You may use the demo code from class as reference to this part of the Lab, create the application so the following list of customers is presented and the Delete link works for each Customer. The add form should always be present unless you are editing, in which case the edit form should always be present, and the message should come up as to which customer was added.

Requirements

1. You must use a **database called "Lab08"**, your username should be 'root' and your password should be '' just like the lab computers.
2. You must use the PDO class we created based on the template provided.
3. You must use classes you should not alter the PDOAgent class we have created.
4. You must list all the customers in the database and provide the provisions to add and delete Customers
5. By default the Add form should be displayed, if the user clicks on the "Edit" action or link you must allow them to edit the entry they clicked on.
6. You may only use one controller file.
7. You must employ a static "DAO" class to support your CRUD operations. Your Mapper class must use the Customer Class. You may not use any array elements that are not objects.
8. As you are using a previous template you must pull out any code that is not used such as the BookMapper and the Book class.
9. You must validate all your input!

Structure

FileName	Description
inc/config.inc.php	Database configuration information
inc/Utility/PDOAgent.class.php	PDO Wrapper Class
inc/Utility/Page.class.php	Page Class
inc/Utility/CustomerDAO.class.php	Static Customer Mapper responsible for CRUD operations

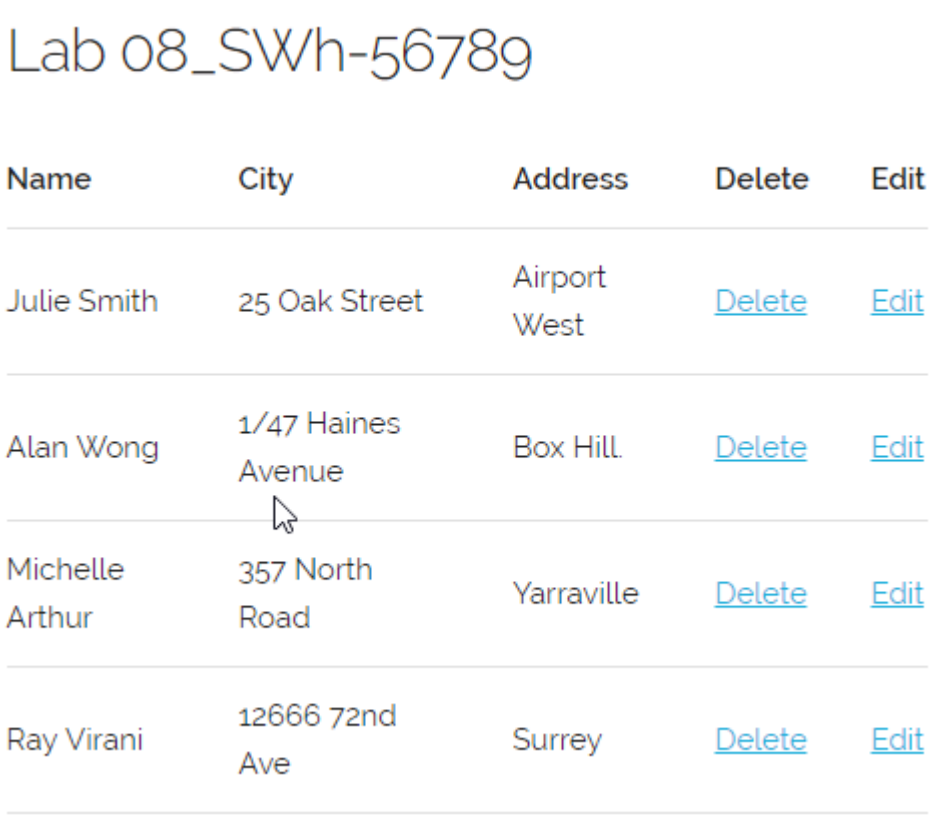
FileName	Description
inc/Entities/Customer.class.php	Customer Entity with getters and setters.
Lab08_SWh-56789.php	The controller class for your application, should be used to support All the CRUD operations

Hints:

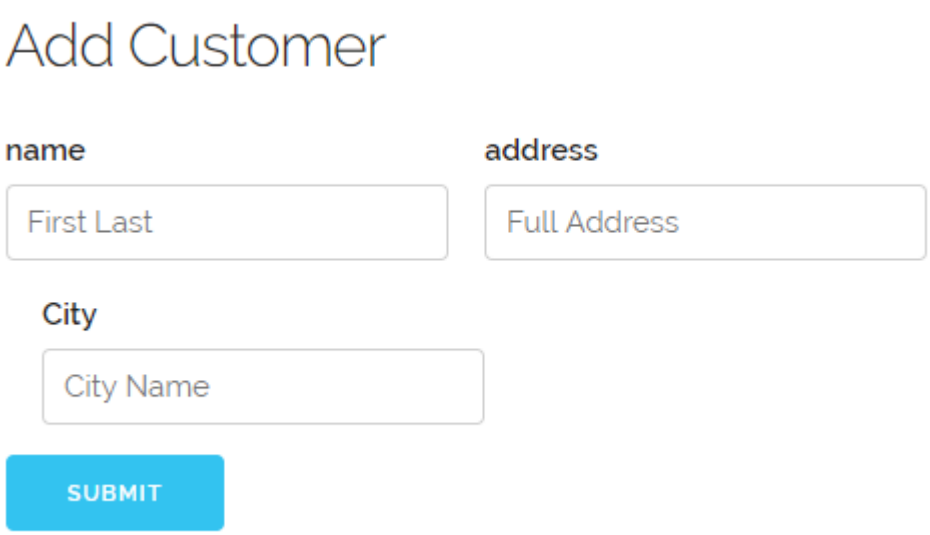
- Use the hidden input type to post hidden data such as the id.
- Use the hidden input type to specify which action you are doing.
- There are alot of moving parts in this assignment **USE XDEBUG** it will save you time!

Appendix:

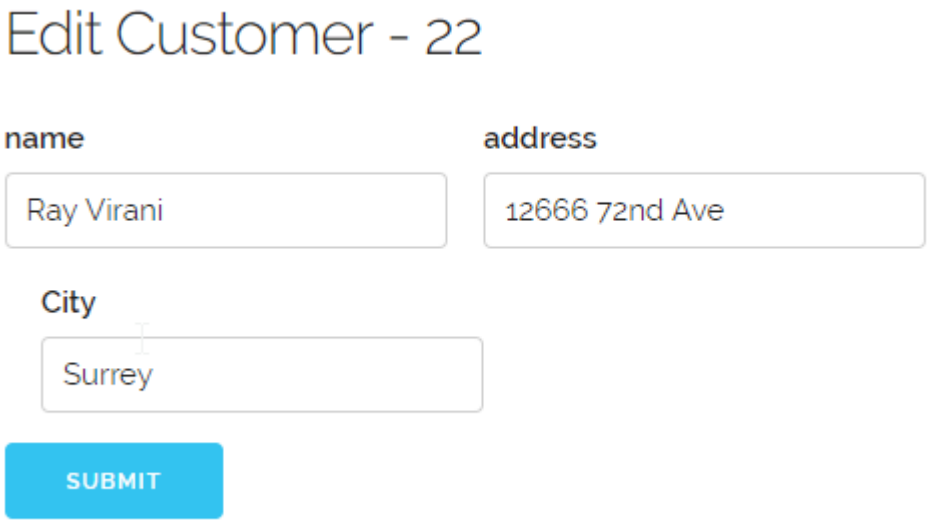
1. List



2. Add Form



3. Edit Form



STOP! - This is a pre-assignment submission checklist!

- Did you remove any **debug output such as var_dump**
- Is your submission clear of any **ORANGE** output from XDebug (Warnings, Info or Errors)?
- Did you follow the naming convention for your files?!
- Did you follow the naming convention for your folder?!
- Does your submission work on a lab computer?!
- Double check **before** submitting

References:

- https://www.tutorialspoint.com/design_pattern/data_access_object_pattern.htm
- Database from PHP and MySQL Web Development (5th Edition)
