Chapter 13 Project 3: CRM Admin

Chapter12-project03.php

1. Two text files containing customer information and their orders
   1. customer.txt
   2. order.txt
2. Define and encapsulate these data in the following order
   1. orderID, customerID, bookISBN, bookTitle, bookCategory
3. Read the data in customer.txt and for each line in customer.txt file, create a new instance of Customer in an array, then display the customer data in a table.
4. Each customer name must be a link back to chapter13-project03.php, but with the customer id data as a query string.
5. When the user clicks on the customer name, then read the data in order.txt

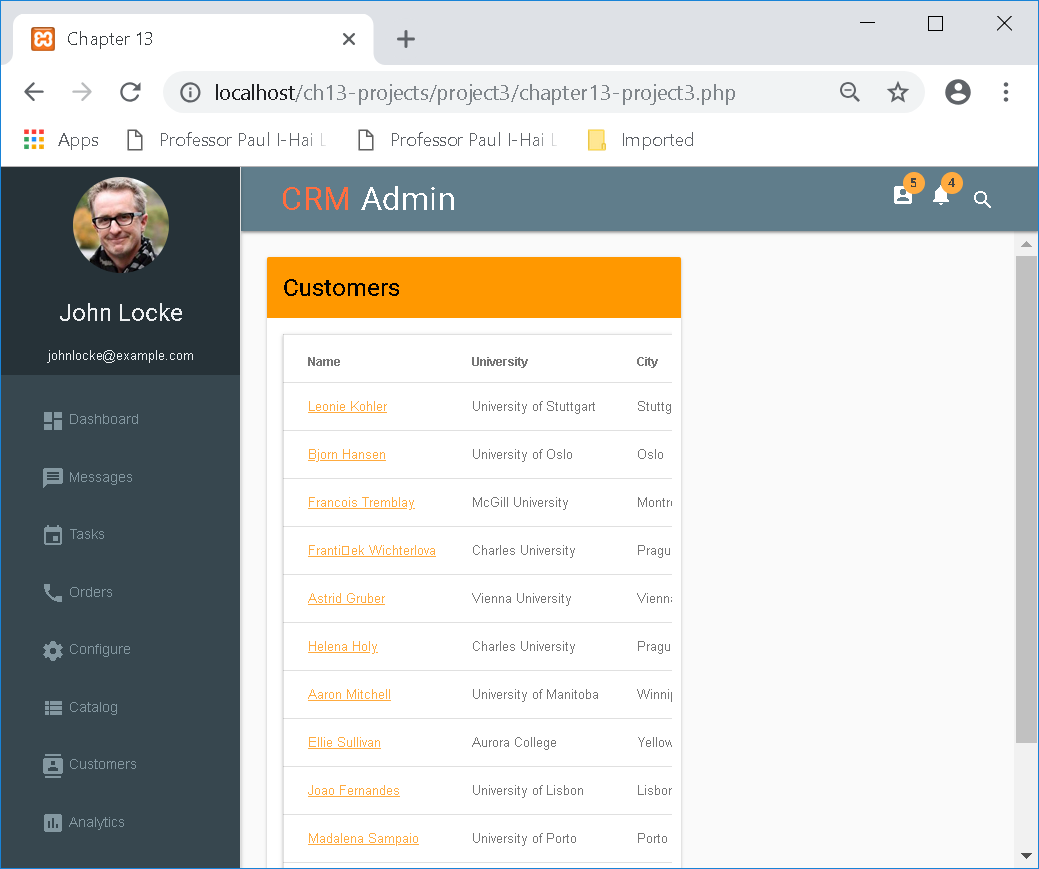
Into an array of Order objects, then display any matching order data for the customer.

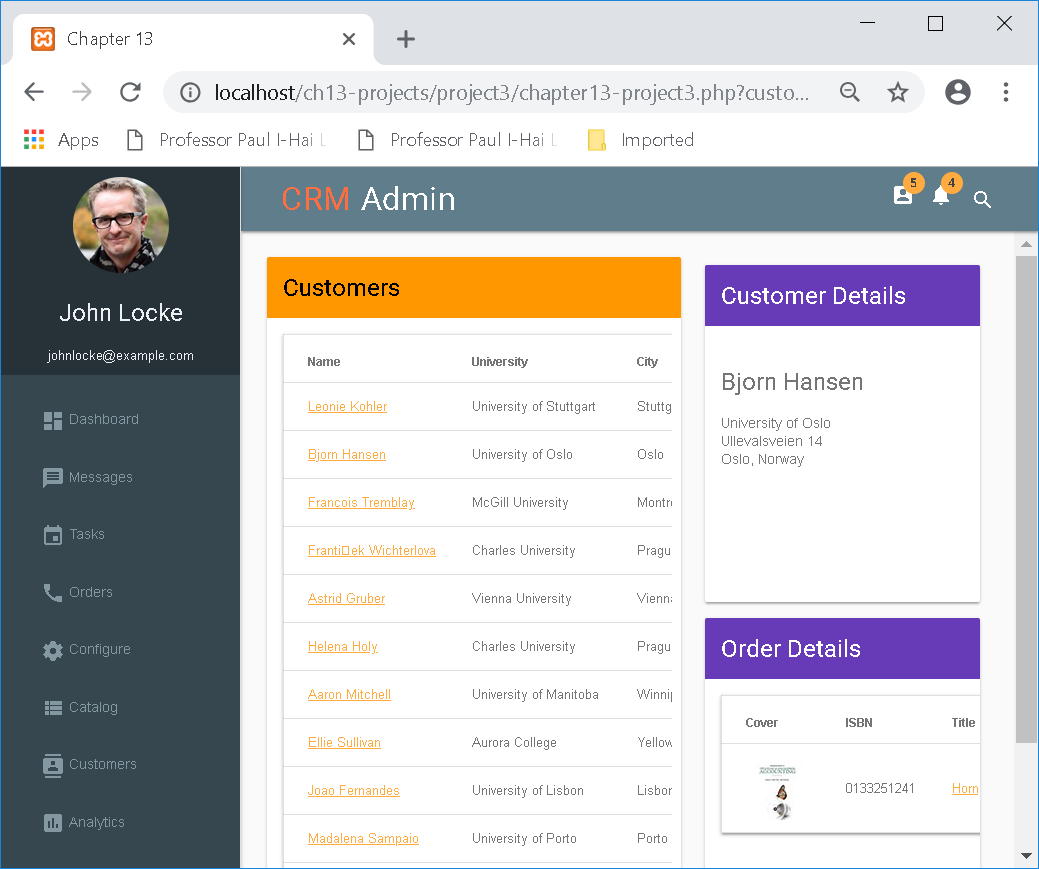
Display a message when there is no order information for the requested customer.

Program Organization

project3 folder

* php.ini --- initialization file for configuring working environment
* css
* data (flat files)
  + books.txt
  + customers.txt
  + orders.txt
* Images
* chapter13-project3.php
  + ‘include/book-utilities.inc.php’
    - Customer.class.php
    - Order.class.php
  + header.inc.php
  + left-nav.inc.php
  + User defined functions
    - function readCustomers($filename)
    - function readOrders($customer, $filename)
* js





<?php

/\*

Represents a Customer (for the book case)

\*/

//Customer.class.php

class Customer

{

private $id;

private $firstName;

private $lastName;

private $email;

private $university;

private $address;

private $city;

private $state;

private $country;

private $zipPostal;

private $phone;

private $sales;

// constructor is

/\*

function \_\_construct($id, $firstName, $lastName, $email, $university, $address, $city, $state, $country, $zip, $phone) {

$this->id = $id;

$this->firstName = $firstName;

$this->lastName = $lastName;

$this->email = $email;

$this->university = $university;

$this->address = $address;

$this->city = $city;

$this->state = $state;

$this->country = $country;

$this->zipPostal = $zip;

$this->phone = $phone;

}

\*/

function \_\_construct($delimiter,$line) {

$splitcontents = explode($delimiter, $line);

$this->id = $splitcontents[0];

$this->firstName = utf8\_encode($splitcontents[1]);

$this->lastName = utf8\_encode($splitcontents[2]);

$this->email = utf8\_encode($splitcontents[3]);

$this->university = utf8\_encode($splitcontents[4]);

$this->address = utf8\_encode($splitcontents[5]);

$this->city = utf8\_encode($splitcontents[6]);

$this->state = utf8\_encode($splitcontents[7]);

$this->country = utf8\_encode($splitcontents[8]);

$this->zipPostal = utf8\_encode($splitcontents[9]);

$this->phone = utf8\_encode($splitcontents[10]);

$this->sales = utf8\_encode($splitcontents[11]);

}

public function getId() { return $this->id; }

public function getFirstName() { return $this->firstName; }

public function getLastName() { return $this->lastName; }

public function getFullName() { return $this->firstName . ' ' . $this->lastName; }

public function getEmail() { return $this->email; }

public function getUniversity() { return $this->university; }

public function getAddress() { return $this->address; }

public function getCity() { return $this->city; }

public function getState() { return $this->state; }

public function getCountry() { return $this->country; }

public function getZipPostal() { return $this->zipPostal; }

public function getPhone() { return $this->phone; }

public function getSales() { return $this->sales; }

}

?>

<?php

/\*

Represents a Order (for the book case)

\*/

//Order.class.php

class Order

{

private $id;

private $customerId;

private $isbn;

private $title;

private $category;

// constructor is

/\*

function \_\_construct($id, $customerId, $isbn, $title, $category) {

$this->id = $id;

$this->customerId = $customerId;

$this->isbn = $isbn;

$this->title = $title;

$this->category = $category;

}

\*/

function \_\_construct($delimiter,$line) {

$splitcontents = explode($delimiter, $line);

$this->id = $splitcontents[0];

$this->customerId = $splitcontents[1];

$this->isbn = $splitcontents[2];

$this->title = $splitcontents[3];

$this->category = $splitcontents[4];

}

public function getId() { return $this->id; }

public function setId($id) { $this->id = $id; }

public function getCustomerId() { return $this->customerId; }

public function getIsbn() { return $this->isbn; }

public function getTitle() { return $this->title; }

public function getCategory() { return $this->category; }

}

?>

<?php

//book-utilities.inc.php

include 'Customer.class.php';

include 'Order.class.php';

function readCustomers($filename) {

// read the file into memory; if there is an error then stop processing

$arr = file($filename) or die('ERROR: Cannot find file');

// our data is semicolon-delimited

$delimiter = ';';

// loop through each line of the file

foreach ($arr as $line) {

// returns an array of strings where each element in the array

// corresponds to each substring between the delimiters

$aCustomer = new Customer($delimiter,$line);

//print\_r($line);

// add customer to array of customers

$customers[$aCustomer->getId()] = $aCustomer;

}

return $customers;

}

function readOrders($customer, $filename) {

$arr = file($filename) or die('ERROR: Cannot find file');

// our data is comma-delimited

$delimiter = ',';

// loop through each line of the file

foreach ($arr as $line) {

$anOrder = new Order($delimiter,$line);

$orders[$anOrder->getId()] = $anOrder;

}

foreach ($orders as $ord) {

if ($ord->getCustomerId() == $customer) $filtered[] = $ord;

}

if (isset($filtered) )

return $filtered;

else

return null;

}

?>

<?php

include 'includes/book-utilities.inc.php';

$customers = readCustomers('data/customers.txt');

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>Chapter 13</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href='http://fonts.googleapis.com/css?family=Roboto' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

<link rel="stylesheet" href="https://code.getmdl.io/1.1.3/material.blue\_grey-orange.min.css">

<link rel="stylesheet" href="css/styles.css">

<script src="https://code.jquery.com/jquery-1.7.2.min.js" ></script>

<script src="https://code.getmdl.io/1.1.3/material.min.js"></script>

<script src="js/jquery.sparkline.2.1.2.js"></script>

<script type="text/javascript">

$(function() {

/\*\* This code runs when everything has been loaded on the page \*/

/\* Inline sparklines take their values from the contents of the tag \*/

$('.inlinesparkline').sparkline('html', {type: 'bar', barColor: '#6200EA'});

});

</script>

</head>

<body>

<div class="mdl-layout mdl-js-layout mdl-layout--fixed-drawer

mdl-layout--fixed-header">

<?php include 'includes/header.inc.php'; ?>

<?php include 'includes/left-nav.inc.php'; ?>

<main class="mdl-layout\_\_content mdl-color--grey-50">

<section class="page-content">

<div class="mdl-grid">

<!-- mdl-cell + mdl-card -->

<div class="mdl-cell mdl-cell--7-col card-lesson mdl-card mdl-shadow--2dp">

<div class="mdl-card\_\_title mdl-color--orange">

<h2 class="mdl-card\_\_title-text">Customers</h2>

</div>

<div class="mdl-card\_\_supporting-text">

<table class="mdl-data-table mdl-shadow--2dp">

<thead>

<tr>

<th class="mdl-data-table\_\_cell--non-numeric">Name</th>

<th class="mdl-data-table\_\_cell--non-numeric">University</th>

<th class="mdl-data-table\_\_cell--non-numeric">City</th>

<th>Sales</th>

</tr>

</thead>

<tbody>

<?php

foreach ($customers as $cust) {

echo '<tr>';

echo '<td class="mdl-data-table\_\_cell--non-numeric"><a href="chapter13-project3.php?customer=' . $cust->getId() . '">' . $cust->getFullName() . '</a></td>';

echo '<td class="mdl-data-table\_\_cell--non-numeric">' . $cust->getUniversity() . '</td>';

echo '<td class="mdl-data-table\_\_cell--non-numeric">' . $cust->getCity() . '</td>';

echo '<td><span class="inlinesparkline">' . $cust->getSales() . '</span></td>';

echo '</tr> ';

}

?>

</tbody>

</table>

</div>

</div> <!-- / mdl-cell + mdl-card -->

<div class="mdl-grid mdl-cell--5-col">

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'GET') {

if ( isset($\_GET['customer']) ) {

$requestedCustomer = $customers[$\_GET['customer']];

?>

<!-- mdl-cell + mdl-card -->

<div class="mdl-cell mdl-cell--12-col card-lesson mdl-card mdl-shadow--2dp">

<div class="mdl-card\_\_title mdl-color--deep-purple mdl-color-text--white">

<h2 class="mdl-card\_\_title-text">Customer Details</h2>

</div>

<div class="mdl-card\_\_supporting-text">

<h4><?php echo $requestedCustomer->getFullName(); ?></h4>

<?php echo $requestedCustomer->getUniversity(); ?><br>

<?php echo $requestedCustomer->getAddress(); ?><br>

<?php echo $requestedCustomer->getCity() . ', ' . $requestedCustomer->getCountry(); ?>

</div>

</div> <!-- / mdl-cell + mdl-card -->

<!-- mdl-cell + mdl-card -->

<div class="mdl-cell mdl-cell--12-col card-lesson mdl-card mdl-shadow--2dp">

<div class="mdl-card\_\_title mdl-color--deep-purple mdl-color-text--white">

<h2 class="mdl-card\_\_title-text">Order Details</h2>

</div>

<div class="mdl-card\_\_supporting-text">

<?php

$orders = readOrders( $\_GET['customer'], 'data/orders.txt' );

if ( is\_null($orders) ) {

echo 'No orders for ' . $requestedCustomer->getFullName();

} else {

?>

<table class="mdl-data-table mdl-shadow--2dp">

<thead>

<tr>

<th class="mdl-data-table\_\_cell--non-numeric">Cover</th>

<th class="mdl-data-table\_\_cell--non-numeric">ISBN</th>

<th class="mdl-data-table\_\_cell--non-numeric">Title</th>

</tr>

</thead>

<tbody>

<?php foreach ($orders as $ord) {

echo '<tr>';

echo '<td><img src="images/tinysquare/' . $ord->getIsbn() . '.jpg"></td>';

echo '<td>' . $ord->getIsbn() . '</td>';

echo '<td class="mdl-data-table\_\_cell--non-numeric"><a href="chapter12-project03.php?isbn=' . $ord->getIsbn() . '">' . $ord->getTitle() . '</td>';

echo '</tr> ';

} ?>

</tbody>

</table>

<?php } ?>

</div>

</div> <!-- / mdl-cell + mdl-card -->

<?php

}

} ?>

</div>

</div> <!-- / mdl-grid -->

</section>

</main>

</div> <!-- / mdl-layout -->

</body>

</html>