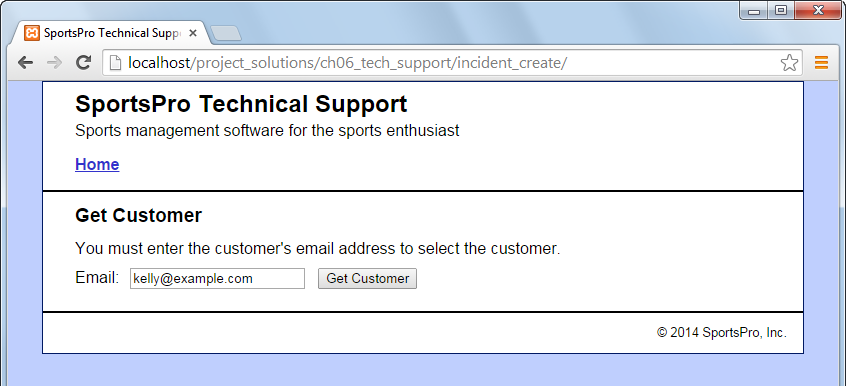
**Assignment 4 – Continuing PHP**

The starting point for the assignment is in your submission for Assignment 3. You will use prepared statements to interact with the database, try/catch blocks to handle any data access errors, and establish a session once you are logged in.

Part F: Create incident (20 points)

For this part, you’ll create an application that lets an admin user enter new incidents. To do that, you’ll begin by letting the user select a customer.

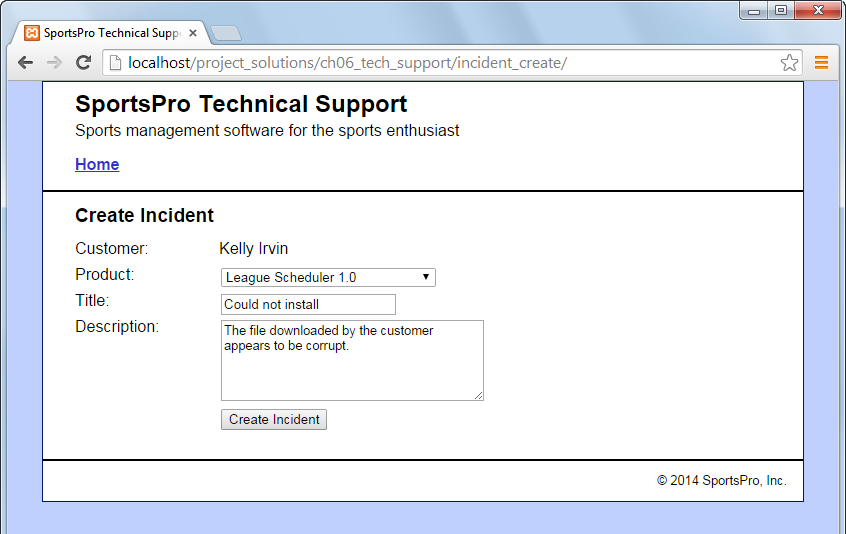
The Get Customer page



Operation

* To get a customer, the user can enter the customer’s email address. Then, the user can click on the Get Customer button to retrieve the customer’s data and display the Create Incident page.
* If the email address is not found in the database, display an error page with the appropriate message.

The Create Incident page (view 1)



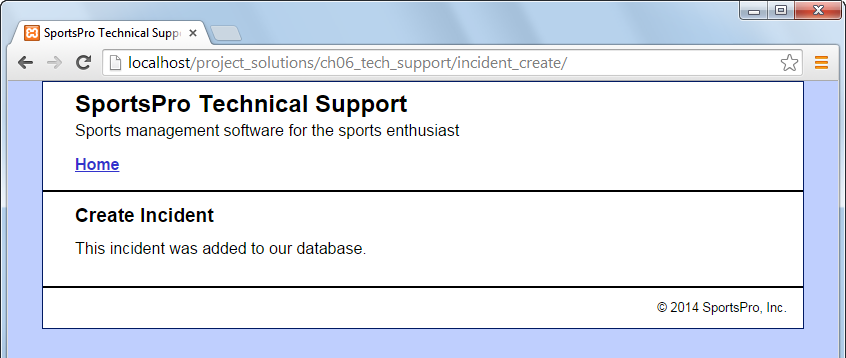
Operation

* To create an incident, the user selects a product from the Product drop-down list, enters a title, enters a description, and clicks on the Create Incident button.

Specifications

* The Product drop-down list should only include products that the customer has registered.
* Hint: If you need help for this drop-down list, see Appendix A & B from Assignment 3.

The Create Incident page (view 2)



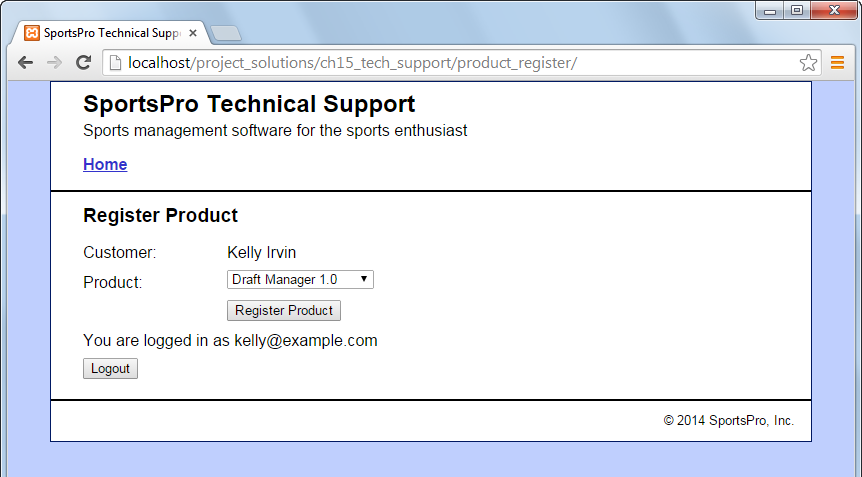
Specifications

* If successful, the Create Incident page should display a message that indicates that the incident was added to the database.

Part G: Use sessions (20 points)

For the Register Product application, let the customer skip the Customer Login page if he or she has already logged in by using a session.

The Register Product page



Operation

* Same as Part D (in Assignment 3), but a customer who has logged in can skip the Customer Login page.
* The customer can view the message on the Register Product page to verify that he or she is logged in.
* To log out, the customer can click on the Logout button or close the browser. If the user clicks the Logout button, the session is destroyed and Customer Login page is displayed.

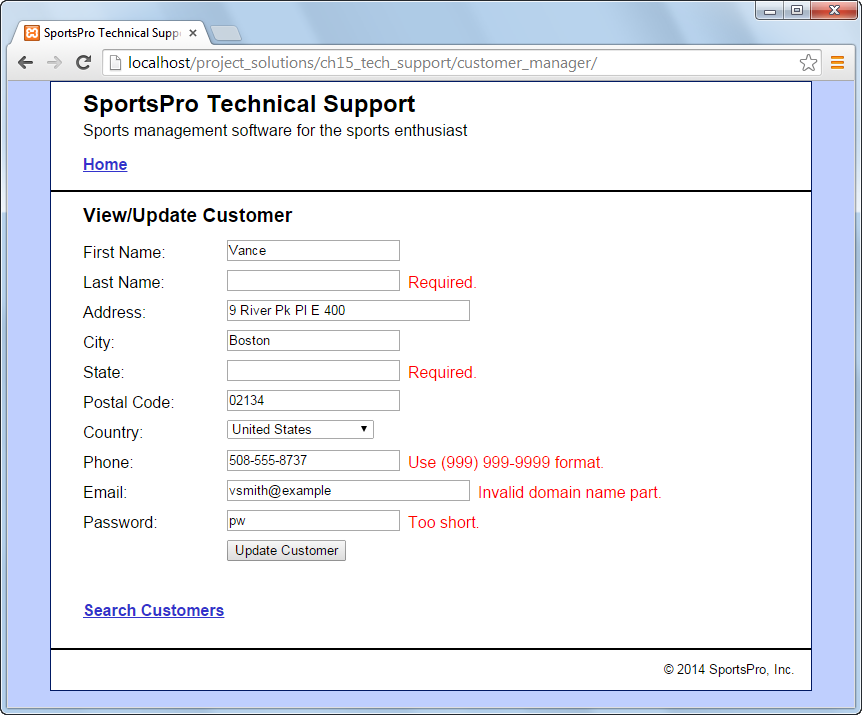
Specifications

* Use $\_SESSION to save all data necessary to maintain the user session.
* Test this functionality using Chrome. You may have to close and reopen the browser between test attempts.

Part H: Improve validation (10 points)

For the Manage Customers application, improve the data validation so it uses the techniques described in class. Review the validation attributes of HTML5 <form> fields to do this part.

The View/Update Customer page



Operation

* Same as Part C in Assignment 3, but with data validation for all fields on the View/Update Customer page.

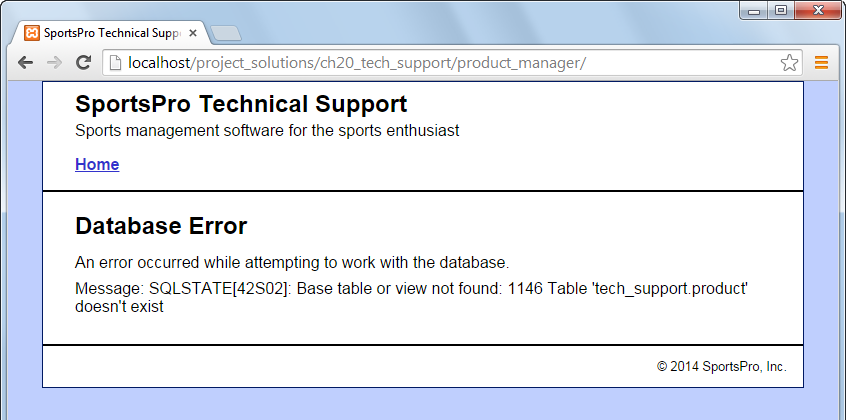
Specifications

* The first name, last name, address, city, and email are required and must have at least 1 and less than 51 characters. The state field is required and must be 2 characters.
* The postal code is required and must have at least 1 and less than 15 characters.
* You don’t need to validate the country field since the drop-down list requires the user to select a valid country.
* The phone number must be in the (999) 999-9999 format.
* The email address must be a valid email address.
* The password must be at least 6 and less than 21 characters.
* All required fields must have the red text “Required” next to the text field

Part I: Handle data access errors (30 points)

For all PHP scripts in the application, use try/catch statements to handle data access errors.

The Database Error page



Operation

* From the user’s point of view, all applications should work the same as they did previously. However, if any application encounters a database error, the user should see a Database Error page like the one above.
* To simulate a database error, you can modify one of the SQL statements in the model so that it causes an error. For example, on the page above, the SQL statement is trying to access the “product” table instead of the “products” table.

Specifications

* Modify all data access functions so they use a try/catch statement to handle any data access errors that may occur. To do that, they can display a Database Error page like the one shown above.
* Make sure all SELECT, INSERT, UPDATE and DELETE that work with data supplied through a form use prepared statements.
* Data from a form is to be checked for HTML injection before entered into the database.
* Be sure that all displayed forms fit on the white (or a color or image of your choice) background so adjust styles of web page objects accordingly.