# An introduction to the coding of the assignment

This introduction describes the design of the SportsPro Technical Support application and the tech\_support database. In addition, it explains how to make the tech\_support database available to your applications, how to restore the database so it contains its original data, and how to prepare for developing the application. Finally, it provides some general information about developing the site.

### The design of the SportsPro Technical Support application

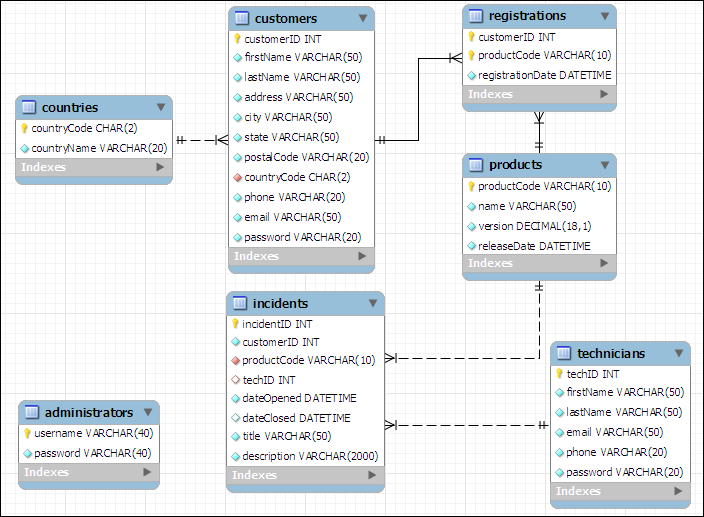
The SportsPro Technical Support application consists of web pages that provide functions for three types of users. First, it lets administrators perform functions such as maintaining the Products, Customers, and Technicians tables of the tech\_support database. Second, it lets technicians perform functions such as updating incidents. And third, it lets customers perform functions such as registering products.

Most steps of this (and future) assignments have you add one or more new pages to the SportsPro application. For example, Part 1 has you add two pages that let an administrator manage the products in the database.

### The design of the Tech Support database

The tech\_support database is used to track technical support incidents. It consists of the seven tables shown in the diagram that follows. The incidents table contains one row for each technical support incident. Each row in the incidents table is related to one row in the customers table, which contains information about the company’s customers; one row in the products table, which contains information about the company’s products; and one row in the technicians table, which contains information about the company’s technical support staff.

In addition, a table named registrations keeps track of the products that are registered to each customer, a table named countries stores the countries of the world, and a table named administrators stores the usernames and passwords for the administrators. Note that the administrators table is not related to any of the other tables.



In addition to the column data types shown above, you should know that the customerID, incidentID, and techID columns in the customers, incidents, and technicians tables are AUTO\_INCREMENT columns. So, the values of these columns are set automatically when new rows are added to these tables. For more details about this database, you can use MySQL Workbench to view the structure and data that’s stored in the database.

## How to install the database

To install the tech\_support database, you can start MySQL Workbench and run the tech\_support.sql script file that’s provided on Blackboard.

### How to restore the database

As you test some of the projects that you develop, you’ll need to add, modify, and delete rows in the database. Then, at some point, you may want to restore the original data. To do that, you can use MySQL Workbench to run the tech\_support.sql file again. This deletes both the structure and the data of the current tech\_support database and restores the original database.

### How to structure your directories

As you develop the web pages needed for each project, you will need to decide where to store the files needed to implement each project. To keep each project independent of other projects, make sure to store each project in its own directory. For example, store the Manage Projects project in its own directory. This directory should have a descriptive name such as manage\_projects or project\_manager.

### How to format the web pages

As you develop the web pages needed for each project, you will need to apply some formatting to them. To make that easier, you can use the main.css file that’s provided in the starter files. If necessary, you can modify this file, but it contains all of the tags needed to format the pages as shown in this document.

## A starting point for the projects

To make it easy to get started, a sample tech\_support directory is posted to Blackboard. This directory includes some of the files for a website that can help you get started with the projects. These files include the tech\_support.sql file that you can use to create the tech\_support database, and a main.css file that you can use to format the web pages. If you run the website, it displays a menu like the one shown here:



Most projects correspond to one of the links on this menu. However, at this point, if you click on any of these links, they display a message that indicates that the page is under construction. That’s because you still need to write the code that implements these projects.

To get started use FileZilla to copy the tech\_support directory onto your website on webdev. It should be at the same directory level as Assign1. You may work directly on webdev to develop the project if you want. However, if you want to start developing on Eclipse first, create an Eclipse PHP Project. Then, copy the files and subdirectories in tech\_support into the new Eclipse PHP project. (Do not copy the tech\_support directory itself.) You’re all set to develop the needed PHP files.

# The parts of the assignment

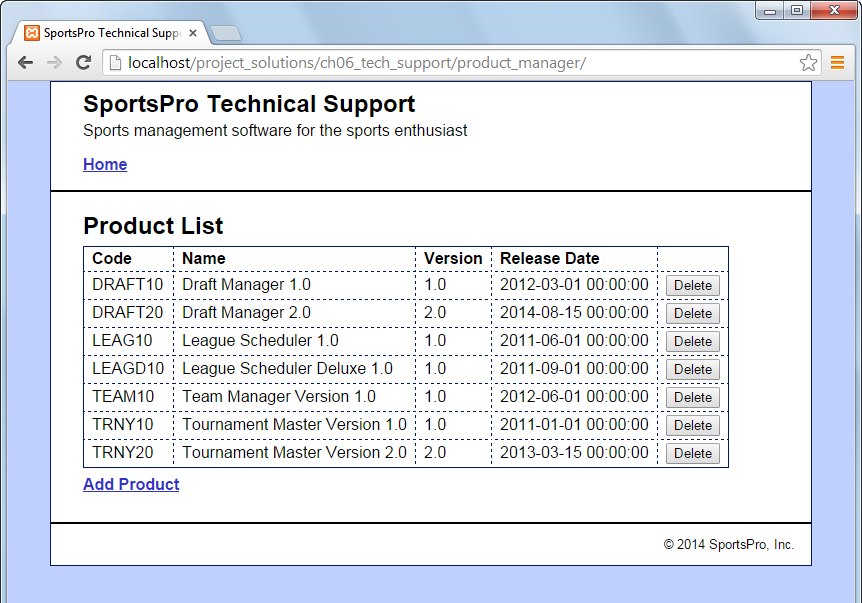
The description of each part of the assignment includes an image of how the pages should appear in a browser, a description of how the pages operate, and specifications for how the assignment should be coded. This information is detailed enough for you to complete each part. However, you’ll need to use your best judgment on how to code many of the details. To do that, write the code in the way that you think is best, based on the skills that are presented in the text and in class.

You can implement each part using any programming techniques you wish. In some cases, however, specifications may direct you to use a specific programming technique. In that case, you should implement the project as directed.

Part A: Manage products (20 points)

For this part, you’ll create an application that lets an admin user view and delete existing products. In addition, this application lets the user add new products by entering the product information into text boxes.

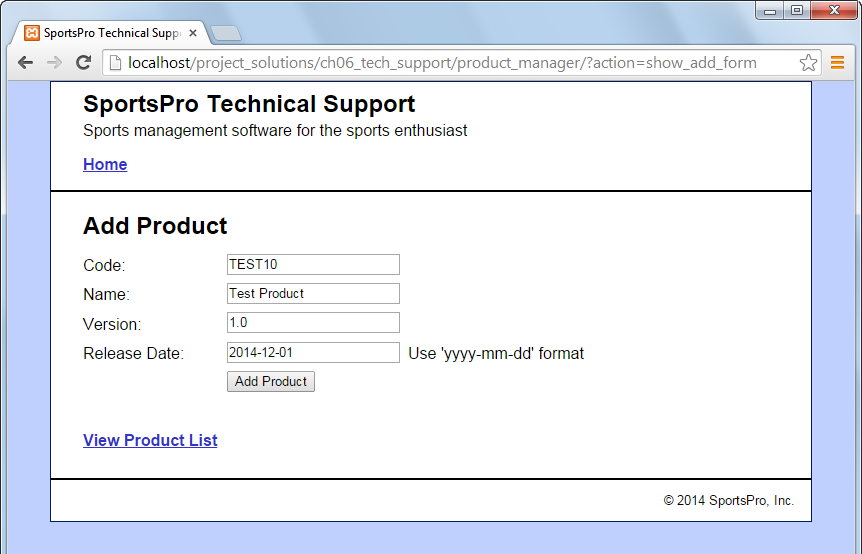
The Product List page



Operation

* When the user clicks the Delete button for a product, the product is deleted from the database.
* When the user clicks the Add Product link, the Add Product page is displayed.
* When the user clicks the Home link, the main menu is displayed.

The Add Product page



Operation

* When the user enters the data for a new product into the text boxes and clicks the Add Product button, the product is added to the database and the Product List page is displayed again, so the user can view the newly added product.
* When the user clicks the View Product List link, the Product List page is displayed.

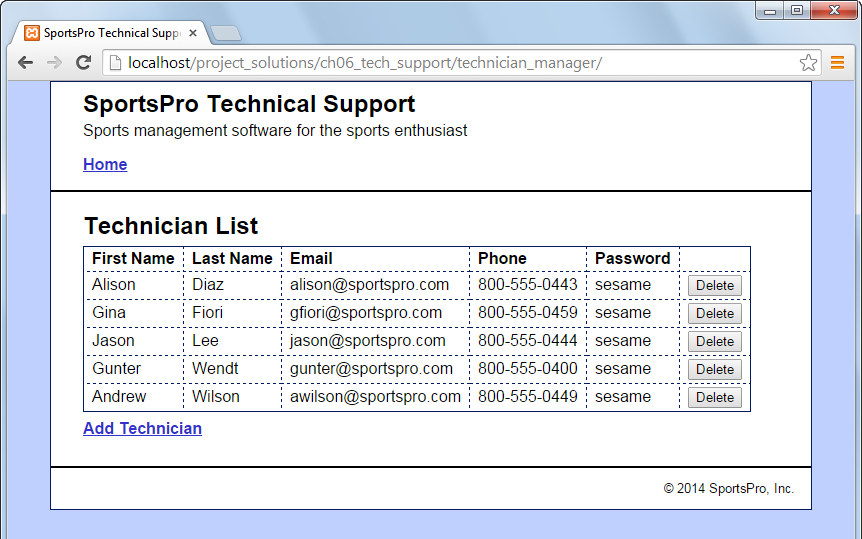
Specifications

* Validate the data the user enters on the Add Product page to be sure that the user enters a product code, name, version, and release date. If this data isn’t provided, display an Error page that indicates that a required field was not entered.
* After the button is clicked, check the database to be sure the record was added.

Part B: Manage technicians (20 points)

For this part, you’ll create an application that lets an admin user view and delete existing technicians. In addition, this application lets the user add a new technician.

The Technician List page



Operation

* When the user clicks the Delete button for a technician, the technician is deleted from the database.
* When the user clicks the Add Technician link, the Add Technician page is displayed.
* Check the database to be sure the record was deleted.

The Add Technician page



Operation

* When the user enters the data for a new technician into the text boxes and clicks the Add Technician button, the technician is added to the database.

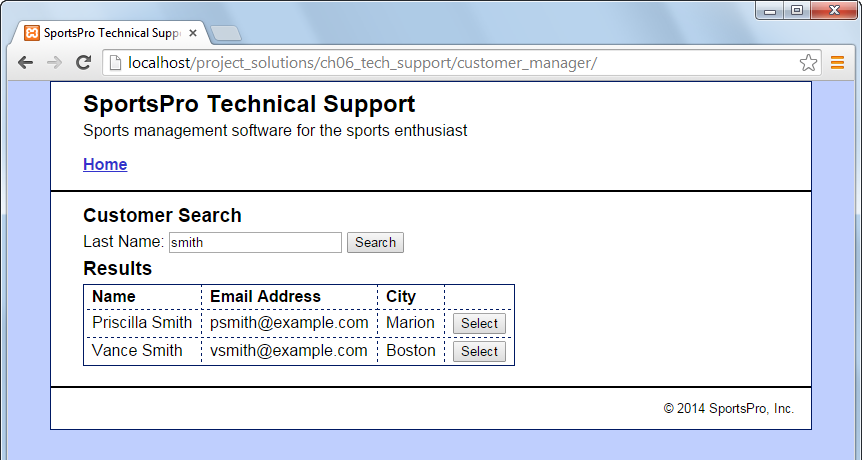
Specifications

* Validate the data the user enters on the Add Technician page to be sure that the user enters data in every text box. If this data isn’t provided, display an Error page that indicates that a required field was not entered.
* After the button is clicked, check the database to be sure the record was added

Part C: Manage customers (20 points)

For this project, you’ll create an application that lets an admin user maintain customer data. To start, this application lets the user select an existing customer. Then, the user can view or update the customer’s data.

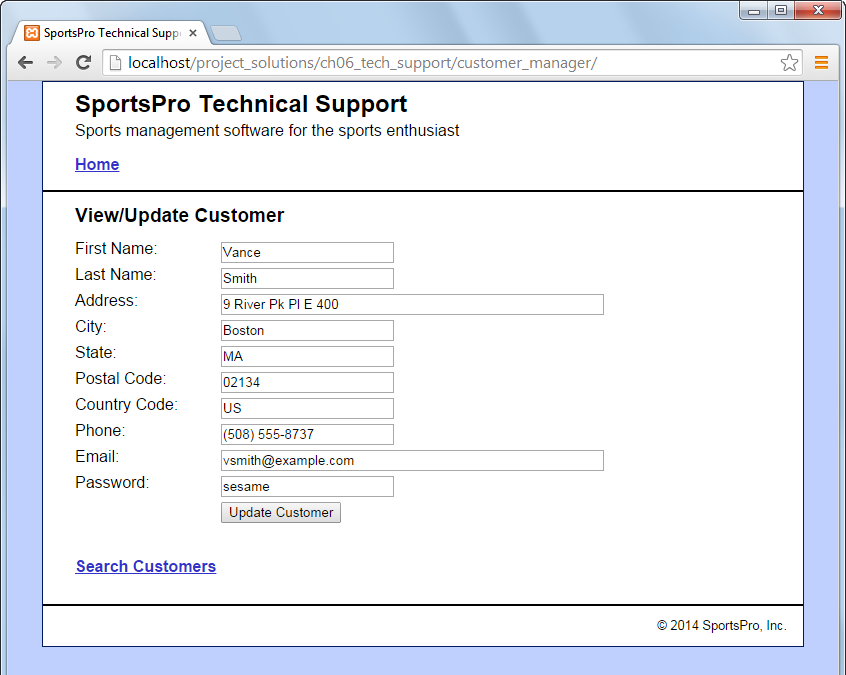
The Select Customer page



Operation

* When the user enters a last name and clicks the Search button, the application displays a table of customers with the specified last name.
* When the user clicks the Select button for a customer, the data for that customer is displayed on the View/Update Customer page.

The View/Update Customer page



Operation

* When the user clicks the Update Customer button for a customer, the application updates the database. The user can also click the Back button or the Search Customers link to return to the Search Customers page without modifying the database.

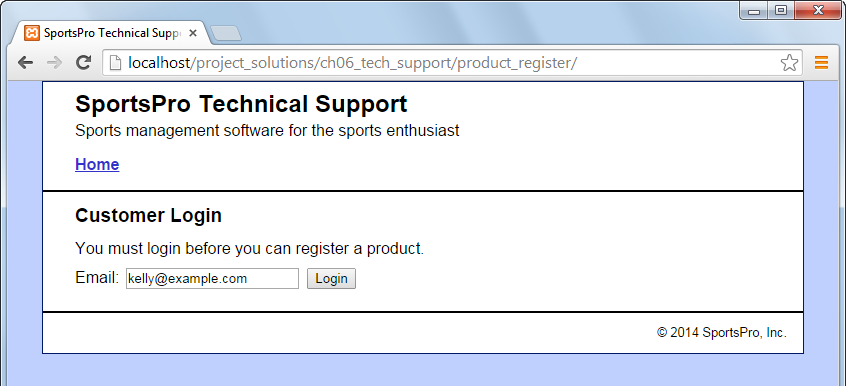
Specifications

* US is the country code for the United States.
* Be sure to check the database to be sure the record was updated.

Part D: Register product (10 points)

For this project, you’ll create an application that lets a customer register a product.

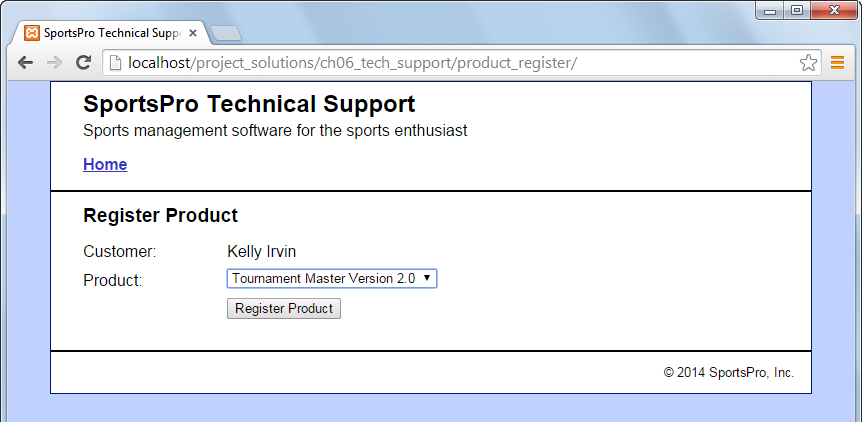
The Customer Login page



Operation

* To log in, the customer can enter his or her email address and click on the Login button.

The Register Product page (view 1)



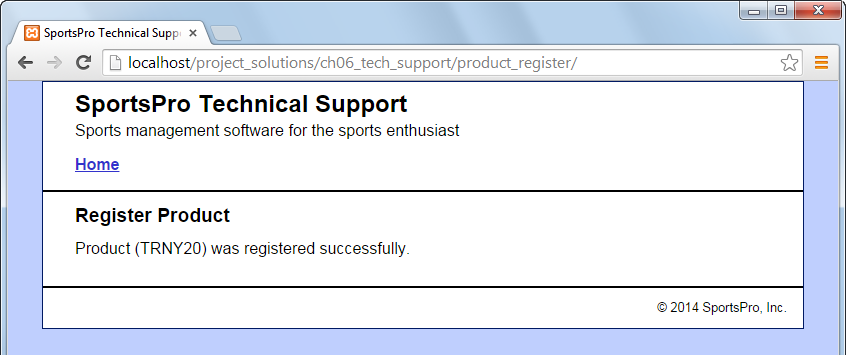
Operation

* To register a product, the customer can select the product and click on the Register Product button.

Specifications

* The Product drop-down list should include all products. If you have any trouble with this, look at Appendix A and Appendix B.

The Register Product page (view 2)



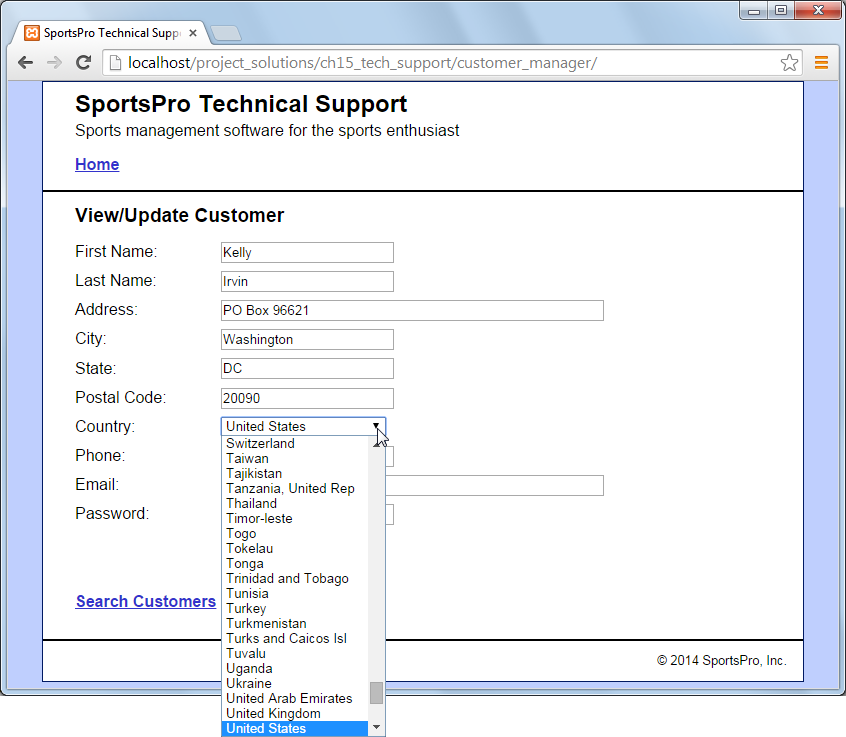
Operation

* After the customer clicks on the Register Product button, the application displays a message that indicates that the product was registered successfully. This message should include the product’s code.
* The registration is recorded in the registrations table.

Part E: Use a drop-down list (10 points)

For this project, you’ll modify the Manager Customers application so it uses a drop-down list to display the country. Also, you’ll allow the user to use this drop-down to change the country. (*see Chapter 7*).

The View/Update Customer page



Operation

* When the user selects a customer, the View/Update Customer page should display the country in a drop-down list, and it should select the correct country for the customer. If you have any trouble with this, look at Appendix B.
* The user can use the Country drop-down list to change the country for the customer.

Specifications

* In the Country drop-down list, display all countries that are available in the countries table in the tech\_support database.
* When the page is first displayed, make sure to select the correct country for the specified customer. To do that, write code that sets the selected attribute of the <option> tag for the appropriate country.

Part F: Website Directory Structure (20 points)

Your website directory structure will be important in the overall Term Project.

The directory structure of your top level website should be as below.

Graphical user interface, text

Description automatically generated with medium confidence

The structure of the tech\_support directory should be similar to what’s below. The webpages you constructed should be in an appropriate directory.

Table

Description automatically generated

Test and Submit Assignment

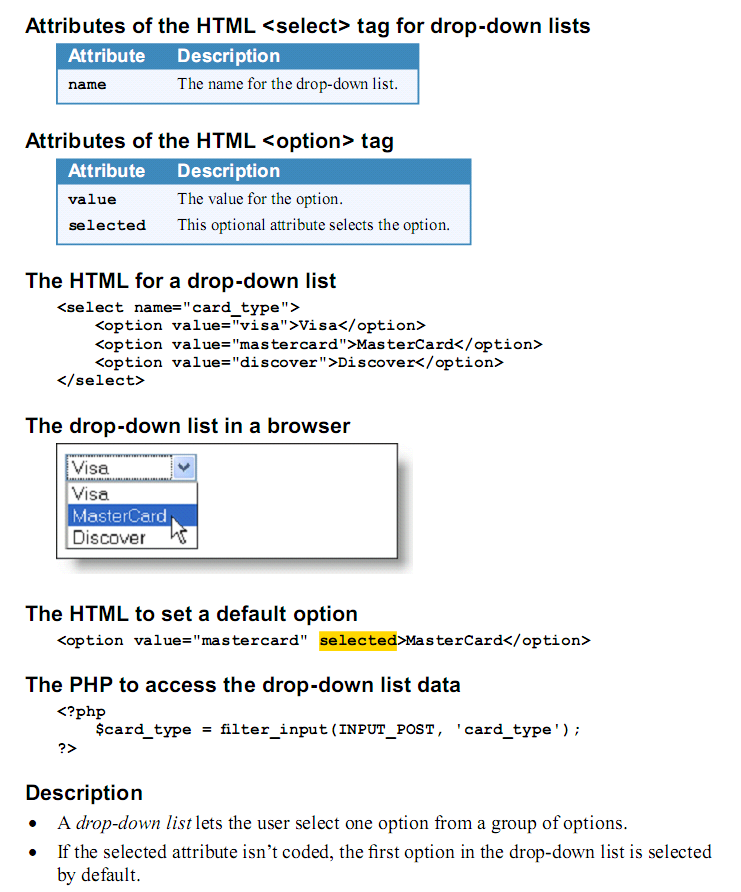
Be sure to thoroughly test your assignment before submitting.

To submit: FileZilla all of your tech\_support files and directories to your webdev directory.

Put a link in your index.html to the tech\_support directory.

It would be a very good idea for each of the project members to test each other’s websites. Let your partners know about any errors that you encounter so they can be fixed.

**APPENDIX A**



**APPENDIX B**

