This application is designed for the technical support department of a hypothetical software company that develops software for sports leagues, and it uses a database named tech\_support.

The purpose of the application is to track technical support service calls (referred to as *incidents*) in a database that also stores information about the company’s customers, software products, and technicians.

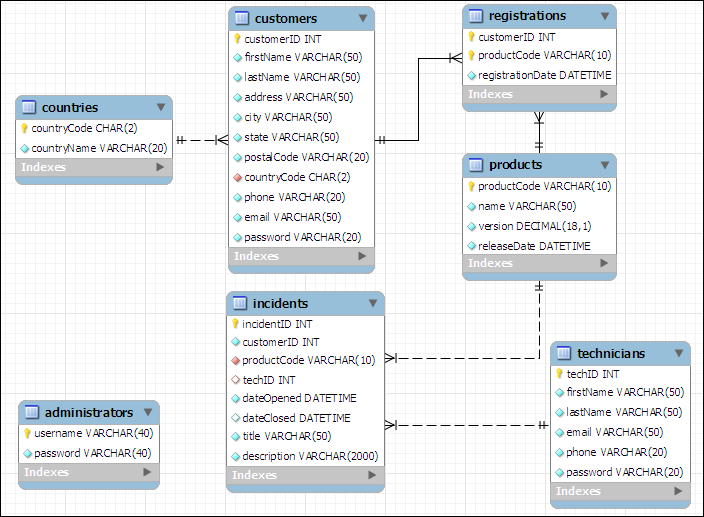
The project contains two parts. The first part of the project.

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 projects.

## 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 phpMyAdmin 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 phpMyAdmin or MySQL Workbench and run the tech\_support.sql file that’s provided.

### 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 phpMyAdmin 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 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 by your instructor. If necessary, you can modify this file, but it contains all of the tags needed to format the pages as shown in this document.

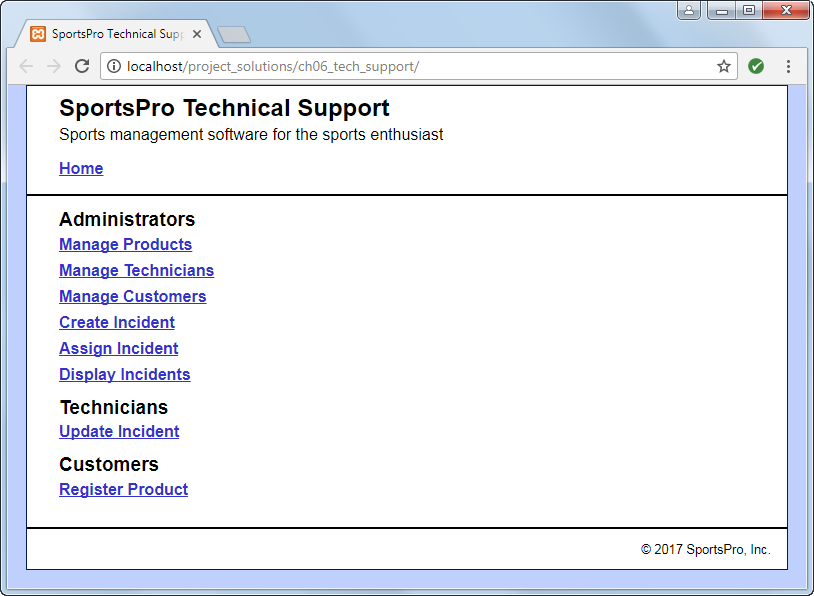
### What to create?

You must create 3 modules with MVC pattern (marked as Mandatory)

You can create the other two modules with MVC pattern to get the bonus points (marked as Optional/Bonus)

## A starting point for the projects

To make it easy to get started, your instructor may give you the tech\_support directory. 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.

# The projects

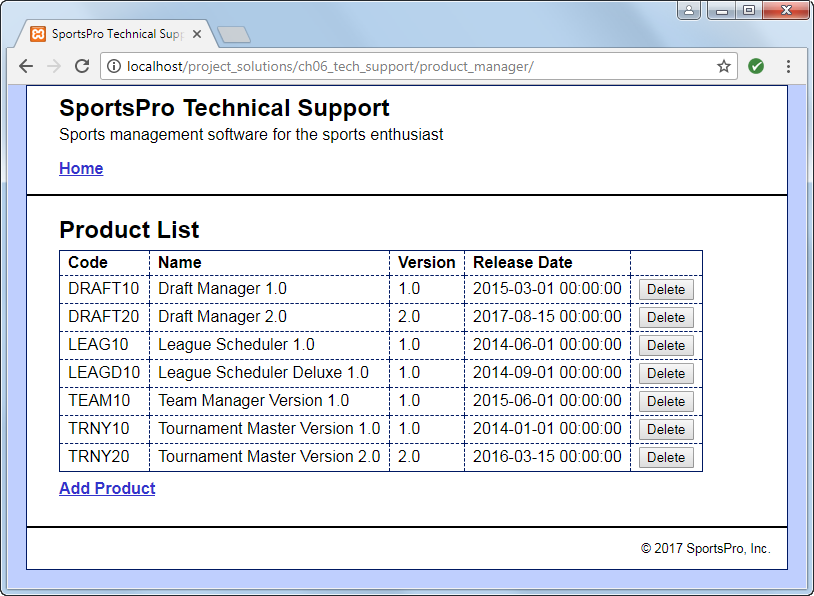
The description of each project includes an image of how the pages should appear in a browser, a description of how the pages operate, and specifications for how the project should be coded. This information is detailed enough for you to complete each project. However, you’ll need to use your best judgment 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 were presented in the book.

Unless you’re instructed otherwise, you can implement each project using any programming techniques you wish. In some cases, however, the project’s specifications will direct you to use a specific programming technique. For example, a project may direct you to use sessions. In that case, you should implement the project as directed.

1. Manage products (10 Pts, Mandatory)

For this module, 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. (*Required reading: chapters 1-6*)

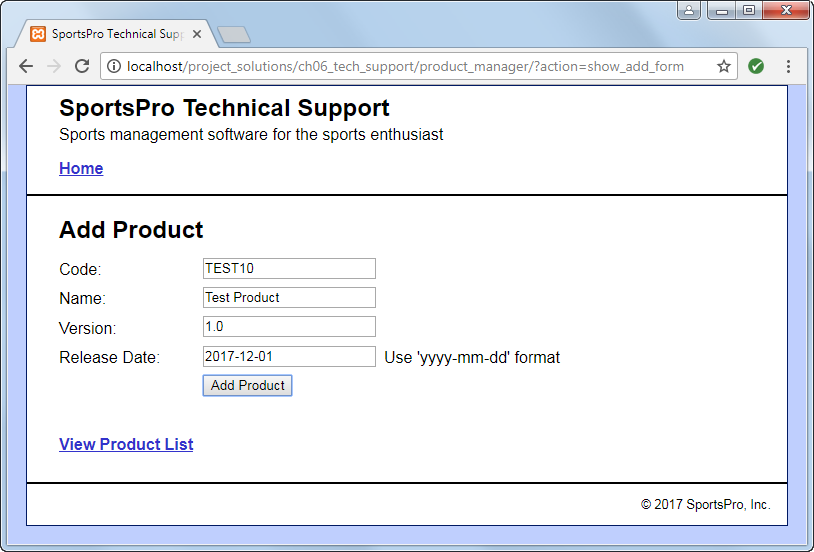
The Product List page



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.

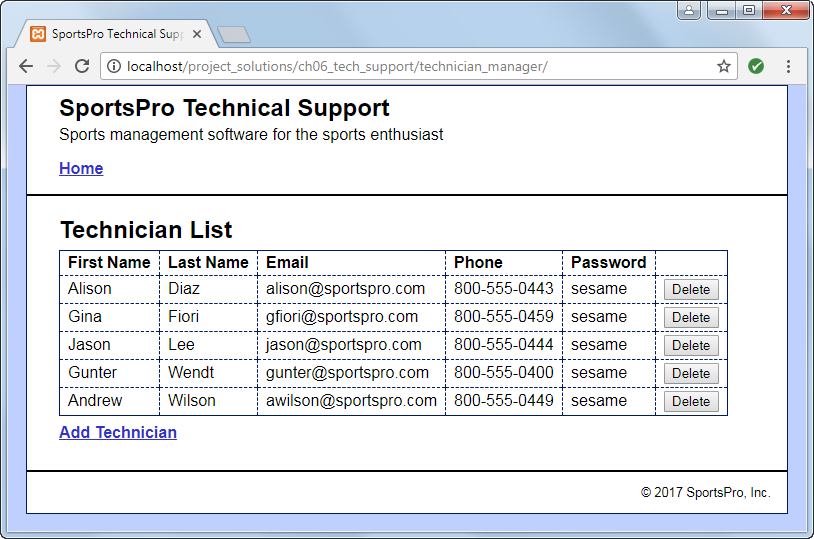
Specifications

* 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.

1. Manage technicians (10 Pts, Optional/Bonus)

For this module, 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. (*Required reading: chapters 1-6*)

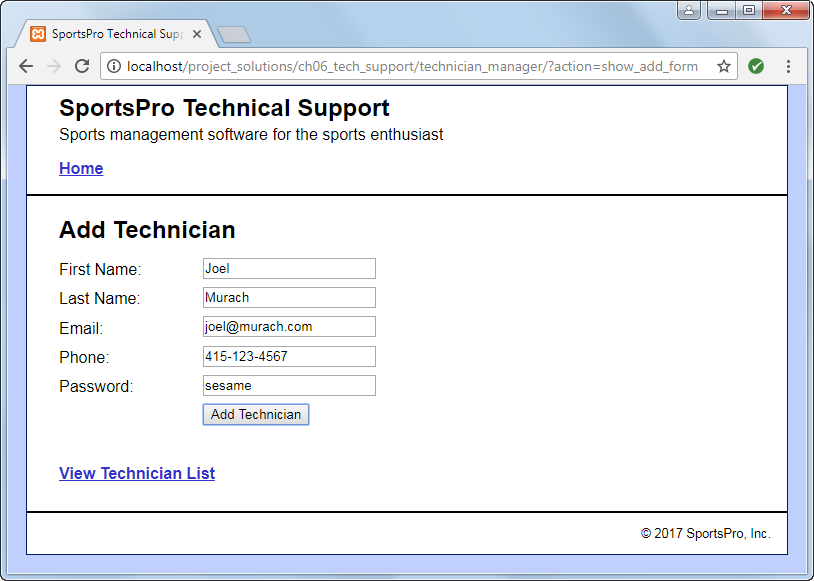
The Technician List page



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.

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.

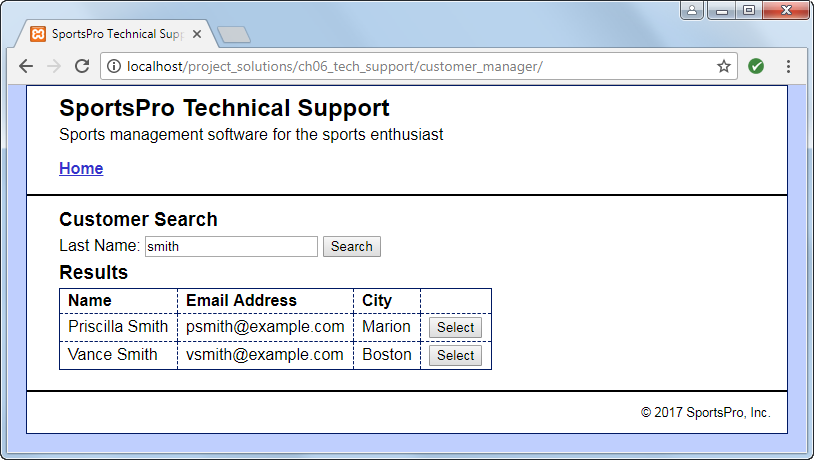
Specifications

* 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.

1. Manage customers (10 Pts, Optional/Bonus)

For this module, 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. (*Required reading: chapters 1-6*)

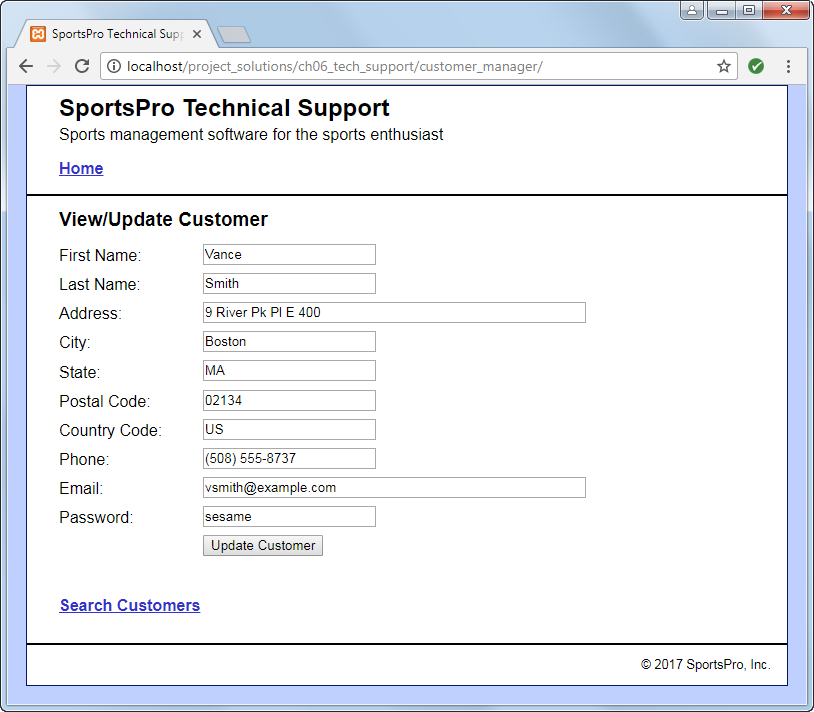
The Select Customer page



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.

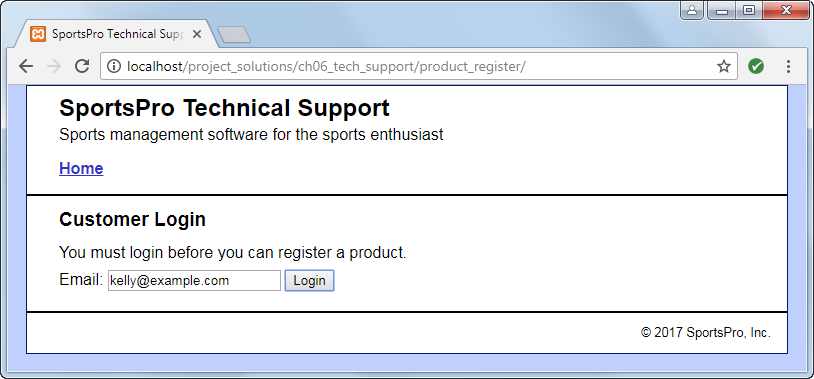
Specifications

* US is the country code for the United States.

1. Register product (10 Pts, Mandatory)

For this module, you’ll create an application that lets a customer register a product. (*Required reading: chapters 1-6*)

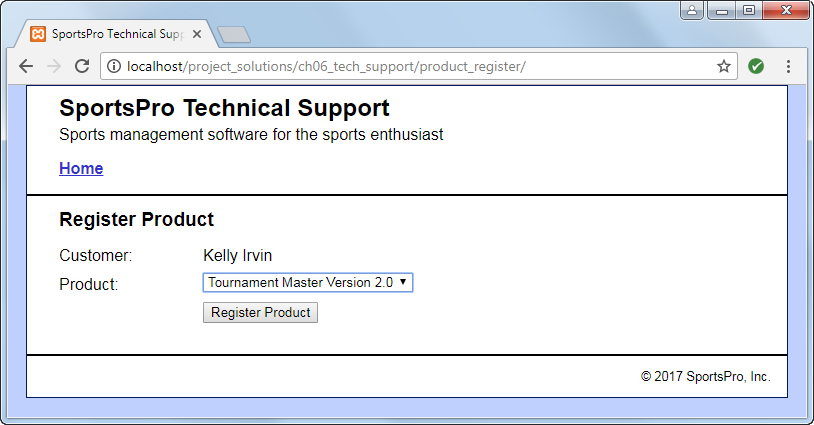
The Customer Login page



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)



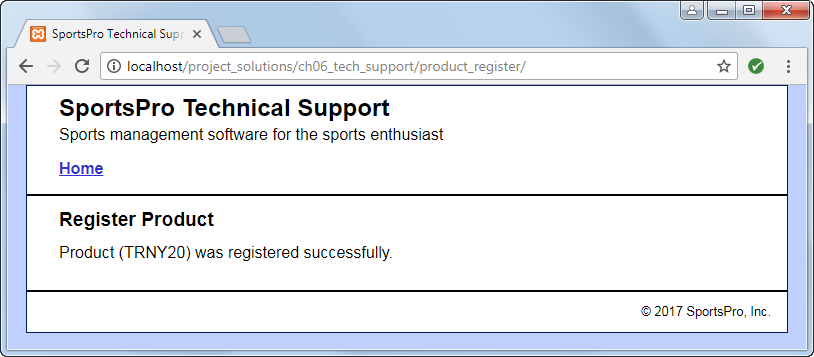
Operation

* 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 ahead to figure 7-5 in chapter 7 and figure 8-11 in chapter 8.

The Register Product page (view 2)



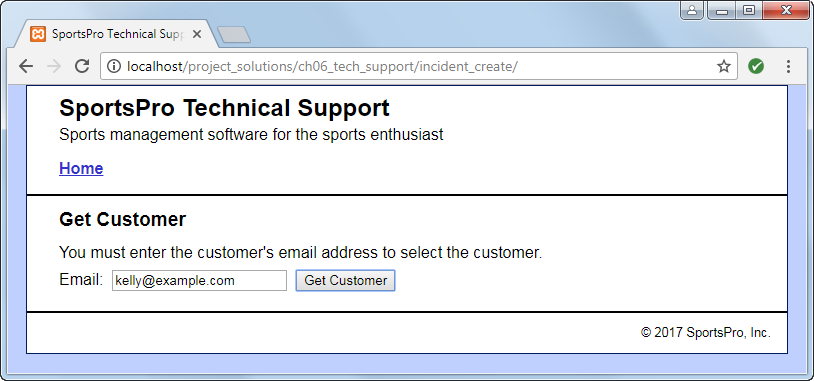
Operation

* 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.

1. Create incident (10 Pts, Mandatory)

For this module, 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. (*Required reading: chapters 1-6*)

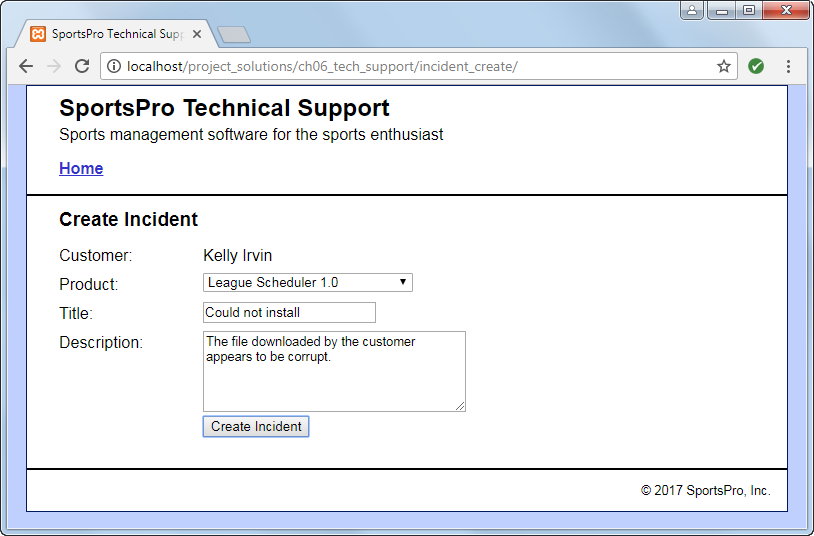
The Get Customer page



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.

The Create Incident page (view 1)



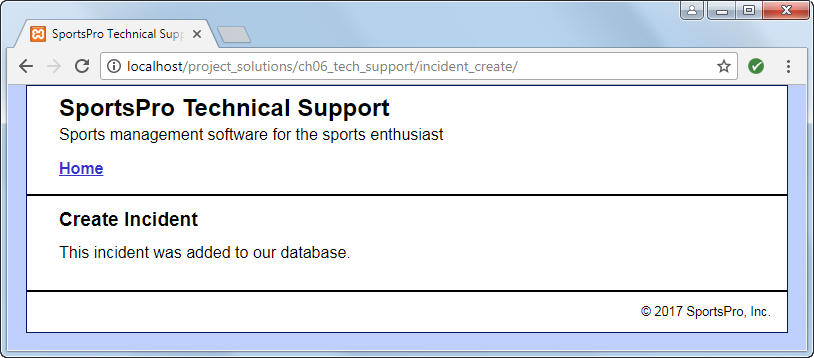
Operation

* 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. If you have any trouble with this drop-down list, look ahead to figure 7-5 in chapter 7 and figure 8-11 in chapter 8.

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.