go deploy ☰

Screenshots ▾

Module 9: Client-Side Development

Lab: Client-Side Development

Scenario

You have been asked to create a web-based ice cream application for your organization's customers. The application should have a page showing all kinds of ice creams in stock, and a purchase page which will allow customers to purchase ice cream. To style the application, you decided to use Bootstrap and Sass. You have decided to use gulp to compile, minify and bundle files.

Exercise 1: Using gulp to Run Tasks

Scenario

In this exercise, you will first install gulp by using npm. You will then create a JavaScript file named gulpfile.js. After that, you will write tasks in the gulpfile.js file to bundle and minify JavaScript files. Finally, you will write a watcher task to track for any changes occurring in files which are located in the Scripts folder.

The main tasks for this exercise are as follows:

- Use npm to install gulp
- Write a task to copy a JavaScript file
- Run the task
- Write a task to minify a JavaScript file
- Write a task to bundle and minify all JavaScript files in a folder
- Add a watcher task
- Run the tasks

Task 1: Use npm to install gulp

□ 1. In the File Explorer, navigate to **D:\Allfiles\Mod09\Labfiles\01_IceCreamCompany_begin**, and then copy the address in the address bar.

□ 2. Select the **Start** button, and then type cmd.

□ 3. Under **Best match**, right-click **Command Prompt**, and then click **Run as administrator**.

□ 5. In the **Administrator: Command Prompt** window, type the following command, and then press Enter.

```
cd {copied folder path}
```

> ⚠ **Note**: If the *{copied folder path}* is different from the disk drive where the command prompt is located, then you should type *{disk drive}:* before typing the **cd** *{copied folder path}* command.

□ 6. In the **Administrator: Command Prompt** window, type the following command, and then press Enter.

```
npm install
```

> ⚠ **Note**: If warning messages are shown at the command prompt you can ignore them.

□ 7. Close the window.

□ 8. In File Explorer, navigate to **D:\Allfiles\Mod09\Labfiles\01_IceCreamCompany_begin**, and then double-click **IceCreamCompany.sln**.

> ⚠ **Note**: If a **Security Warning for IceCreamCompany** dialog box appears, verify that the **Ask me for every project in this solution** check box is cleared, and then click OK.

□ 9. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **Tools** menu, click **Options**.

□ 10. In the **Options** dialog box, in the **Search Options** box, type **Web Package Management**, and then press Enter.

□ 11. In the **Locations of external tools** list box, select **$(PATH)**, press the Up arrow button until **$(PATH)** is at the top of the list, and then click **OK**.

□ 12. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, click **package.json**.

□ 13. Examine the **package.json** code window.

uglify packages appear in the **devDependencies** section and **bootstrap**, **hoek**, **jquery**, **lodash**, **popper.js** packages appear in the **Dependencies** section.

Task 2: Write a task to copy a JavaScript file

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **IceCreamCompany**, point to **Add**, and then click **New Item...**.

□ 2. In the **Add New Item - IceCreamCompany** dialog box, click **Web**, and then in the result pane, click **JavaScript File**.

□ 3. In the **Add New Item - IceCreamCompany** dialog box, in the **Name** box, type gulpfile, and then click **Add**.

□ 4. In the **gulpfile.js** code window, type the following code:

```
var gulp = require('gulp');

var paths = {
    webroot: "./wwwroot/",
    nodeModules: "./node_modules/"
};

paths.jqueryjs = paths.nodeModules + "jquery/dist/jquery.js";
paths.destinationjsFolder = paths.webroot + "scripts/";

gulp.task("copy-js-file", function() {
    return gulp.src(paths.jqueryjs)
        .pipe(gulp.dest(paths.destinationjsFolder));
});
```

Task 3: Run the task

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **gulpfile.js**, and then click **Task Runner Explorer**.

> ⚠ **Note**: If the **Tasks** list does not contain a task named **copy-js-file**, click **Refresh**.

□ 3. In **Task Runner Explorer**, under **Tasks**, right-click **copy-js-file**, and then click **Run**.

> ℹ **Note**: In Solution Explorer, under **wwwroot**, a new folder has been added named **scripts** with a JavaScript file named **jquery.js**

```
var gulp = require('gulp');
```

☐ 2. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
```

☐ 3. In the **gulpfile.js** code window, locate the following code:

```
paths.destinationjsFolder = paths.webroot + "scripts/";
```

☐ 4. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
paths.vendorjsFileName = "vendor.min.js";
```

☐ 5. In the **gulpfile.js** code window, locate the following code:

```
gulp.task("copy-js-file", function() {
    return gulp.src(paths.jqueryjs)
            .pipe(gulp.dest(paths.destinationjsFolder));
});
```

☐ 6. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

```
gulp.task("min-vendor:js", function() {
    return gulp.src(paths.jqueryjs)
            .pipe(concat(paths.vendorjsFileName))
            .pipe(uglify())
            .pipe(gulp.dest(paths.destinationjsFolder));
});
```

Task 5: Write a task to bundle and minify all JavaScript files in a folder

☐ 1. In the **gulpfile.js** code window, locate the following code:

```
    paths.vendorjsFileName = "vendor.min.js";
```

☐ 2. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
paths.jsFiles = "./Scripts/*.js";
paths.jsFileName = "script.min.js";
```

```
gulp.task("min:js", function() {
    return gulp.src(paths.jsFiles)
        .pipe(concat(paths.jsFileName))
        .pipe(uglify())
        .pipe(gulp.dest(paths.destinationjsFolder));
});
```

Task 6: Add a watcher task

□ 1. In the **gulpfile.js** code window, add the following code to the end of the file:

```
gulp.task("js-watcher", function() {
    gulp.watch('./Scripts/*.js', gulp.series("min:js"));
});
```

Task 7: Run the tasks

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **gulpfile.js**, and then click **Task Runner Explorer**.

> ⚠ **Note**: If the **Tasks** list doesn't contain the **min:js**, **min-vendor:js** and **js-watcher** tasks, click **Refresh**.

□ 3. In **Task Runner Explorer**, under **Tasks**, right-click **min-vendor:js**, and then click **Run**.

> ℹ **Note**: In Solution Explorer, under **wwwroot**, under **scripts**, a new file named **vendor.min.js** has been added. Notice that this file is a minified version of the **jquery.js** file.

□ 4. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, expand **Scripts**, and then click **payment-calc.js**.

> ⚠ **Note**: In the fourth line there is a 'deliberate error': **form-control-mistake**.

□ 5. In **Task Runner Explorer**, under **Tasks**, right-click **min:js**, and then click **Run**.

has been added.

□ 6. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, under **wwwroot**, under **scripts**, click **script.min.js**.

> **Note**: The **script.min.js** file is a minified version of the **payment-calc.js** file. It contains the string **form-control-mistake**.

□ 7. In **Task Runner Explorer**, under **Tasks**, right-click **js-watcher**, and then click **Run**.

□ 8. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, under **Scripts**, click **payment-calc.js**.

□ 9. In the **payment-calc.js** code window, select the following code:

```
$('.form-control-mistake')
```

□ 10. Correct the error by replacing the selected code with the following code:

```
$('.form-control')
```

□ 11. Save the edited **payment-calc.js** file.

> **Note**: You may notice the the **min:js** script runs automatically in the **js-watcher (running)** window. This is because of the dependency relationship between the **payment-calc.js** and **script.min.js** files.

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, under **wwwroot**, under **scripts**, click **script.min.js**.

> **Note**: In the **script.min.js** file, the string **form-control-mistake** has been replaced with **form-control**.

> ✓ **Results**: In this exercise, you used gulp to copy, bundle and minify JavaScript files, and add watcher tasks to keep them up-to-date.

Exercise 2: Styling by Using Sass

In this exercise, you will first create a Sass file named main.scss and fill its content. After that, you will create a gulp task to compile the Sass file to a CSS file. Then you will create a gulp watcher task so compilation of the Sass file to a CSS file will be done automatically when the Sass file is changed.

The main tasks for this exercise are as follows:

- Add a new Sass file to the project
- Add gulp tasks to handle the Sass files
- Run a task

Task 1: Add a new Sass file to the project

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **IceCreamCompany**, point to **Add**, and then click **New Folder**.

□ 2. In the **NewFolder** box, type **Styles**, and then press Enter.

□ 3. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **Styles**, point to **Add**, and then click **New Item...**.

□ 4. In the **Add New Item - IceCreamCompany** dialog box, in the search box, type **scss**, and then press Enter.

□ 5. In the **Add New Item - IceCreamCompany** dialog box, click **SCSS Style Sheet (SASS)**.

□ 6. In the **Add New Item - IceCreamCompany** dialog box, in the **Name** box, type **main**, and then click **Add**.

□ 7. In the **main.scss** code window, select the following code:

```
body {
}
```

□ 8. Replace the selected code with the following code:

```
@mixin normalized-text() {
    font-family: "Playfair Display", Arial, Tahoma, sans-serif;
    text-align: center;
}

@mixin normalized-image() {
    width: 100%;
    height: auto;
}
```

☐ 9. In the **main.scss** code window, append the following code:

```
        @include normalized-text();
        font-size: 45px;
        line-height: 50px;
        font-weight: 400;
        letter-spacing: 1px;
        color: #736454;
        margin: 60px;
    }
}

.main-title {
    background-image: url("/images/banner-1.jpg");
    width: 100%;
    background-size: cover;
    background-position: center center;
    min-height: 400px;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;

    h1 {
        @include normalized-text();
        color: $highlights;
        font-size: 50px;
        text-shadow: 0px 2px 5px #aba8a8;
        font-weight: bolder;
                                                                text-align: center;
    }

    button {
        @include normalized-text();
        transition: none;
        color: lighten(#ffffff,90%);
        text-align: inherit;
        line-height: 13px;
        border: 1px solid #d3c0c0;
        margin: 0px;
        padding: 12px 24px;
        letter-spacing: 0px;
        font-weight: 400;
        font-size: 16px;
        font-weight: bold;
        background-color: #736454;
    }
}
```

□ 10. Place the cursor immediately after the last typed **}** (closing bracket) sign, press Enter, and then type the following code:

```
    flex-wrap: wrap;
    justify-content: space-around;
    align-items: flex-end;

    .item {
        color: white;
        width: 200px;
        display: flex;
        flex-direction: column;
        justify-content: space-between;

        h3 {
            @include normalized-text();
            color: #736454;
            font-size: 20px;
        }

        div {
            img {
                    @include normalized-image();
            }
        }

        div {
            p {
                    @include normalized-text();
                    color: #736454;
                    font-size: 20px;
                    margin: 70px;
            }
        }
    }
}

.container {
    .checkout {
            border: 1px solid #ccc;
            box-shadow: 0 0 5px #ccc;
            padding: 20px;
            width: 800px;
            margin: 0 auto;
            border-radius: 4px;
            background-color: #f9f9f9;

            .row justify-content-center intro-row {
                font-weight: bold;
            }
    }
}
```

```
.justify-content-center {
    justify-content: center !important;
    align-items: center;
}

nav {
    width: 450px;
}

img {
    height: 35px;
    width: 35px;
}

.navbar-nav > li {
    float: left;
    position: relative;
}

.row {
    margin: 10px;
}

.imageDisplay {
    @include normalized-image();
}
```

Task 2: Add gulp tasks to handle Sass files

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, click **gulpfile.js**.

□ 2. In the **gulpfile.js** code window, locate the following code:

```
var gulp = require('gulp');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
```

□ 3. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
var sass = require('gulp-sass')(require('sass'));
var cssmin = require('gulp-cssmin');
```

□ 4. In the **gulpfile.js** code window, locate the following code:

```
paths.jsFileName = "script.min.js";
paths.sassFiles = "./Styles/*.scss";
paths.compiledCssFileName = "main.min.css";
paths.destinationCssFolder = paths.webroot + "css/";
```

□ 6. In the **gulpfile.js** code window, locate the following code:

```
gulp.task("min:js", function() {
    return gulp.src(paths.jsFiles)
            .pipe(concat(paths.jsFileName))
            .pipe(uglify())
            .pipe(gulp.dest(paths.destinationjsFolder));
});
```

□ 7. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

```
gulp.task("min:scss", function() {
    return gulp.src(paths.sassFiles)
        .pipe(sass().on('error', sass.logError))
        .pipe(concat(paths.compiledCssFileName))
        .pipe(cssmin())
        .pipe(gulp.dest(paths.destinationCssFolder));
});
```

□ 8. In the **gulpfile.js** code window, locate the following code:

```
gulp.task("js-watcher", function() {
    gulp.watch('./Scripts/*.js', gulp.series("min:js"));
});
```

□ 9. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

```
gulp.task("sass-watcher", function() {
    gulp.watch('./Styles/*.scss', gulp.series("min:scss"));
});
```

Task 3: Run the tasks

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, right-click **gulpfile.js**, and then click **Task Runner Explorer**.

Refresh.

□ 3. In **Task Runner Explorer**, under **Tasks**, right-click **min:scss**, and then click **Run**.

> **Note**: In Solution Explorer, under **wwwroot**, under **css**, a new css file has been added named **main.min.css**.

□ 4. In **Task Runner Explorer**, under **Tasks**, right-click **sass-watcher**, and then click **Run**.

> **Note**: From now whenever you change the **main.scss** file, the **main.min.css** file will automatically be changed.

> ✓ **Results**: In this exercise, you created Sass files and add gulp tasks to compile, bundle and minify them.

Exercise 3: Using Bootstrap

Scenario

In this exercise, you will first update the min-vendor:js task that bundles and minifies JavaScript files to include the JavaScript files of Bootstrap. After that, you will add a task to handle the CSS files of Bootstrap. You will then run the tasks to create the vendor.min.css file and to update the vendor.min.js file. After that, you will style the layout by using Bootstrap. Finally, you will create a view to buy an ice cream and style it by using Bootstrap.

The main tasks for this exercise are as follows:

- Update gulpfile.js to handle Bootstrap
- Run the tasks
- Style the application by using Bootstrap
- Run the application

Task 1: Update gulpfile.js to handle Bootstrap

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, click **gulpfile.js**.

□ 2. In the **gulpfile.js** code window, locate the following code:

```
paths.jqueryjs = paths.nodeModules + "jquery/dist/jquery.js";
paths.popperjs = paths.nodeModules + "popper.js/dist/umd/popper.js";
paths.bootstrapjs = paths.nodeModules + "bootstrap/dist/js/bootstrap.js";
paths.vendorjs = [paths.jqueryjs, paths.popperjs, paths.bootstrapjs];
```

□ 4. In the **gulpfile.js** code window, select the following code:

```
gulp.task("min-vendor:js", function() {
    return gulp.src(paths.jqueryjs)
```

□ 5. Replace the selected code with the following code:

```
gulp.task("min-vendor:js", function() {
    return gulp.src(paths.vendorjs)
```

□ 6. In the **gulpfile.js** code window, locate the following code:

```
paths.destinationCssFolder = paths.webroot + "css/";
```

□ 7. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
paths.bootstrapCss = paths.nodeModules + "bootstrap/dist/css/bootstrap.css";
paths.vendorCssFileName = "vendor.min.css";
```

□ 8. In the **gulpfile.js** code window, locate the following code:

```
gulp.task("min:scss", function() {
    return gulp.src(paths.sassFiles)
            .pipe(sass().on('error', sass.logError))
            .pipe(concat(paths.compiledCssFileName))
            .pipe(cssmin())
            .pipe(gulp.dest(paths.destinationCssFolder));
});
```

□ 9. Place the cursor at the end of the located code, press Enter twice, and then type the following code:

```
gulp.task("min-vendor:css", function() {
    return gulp.src(paths.bootstrapCss)
            .pipe(concat(paths.vendorCssFileName))
            .pipe(cssmin())
            .pipe(gulp.dest(paths.destinationCssFolder));
});
```

Task 2: Run the tasks

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

> ⚠ **Note**: If the **Tasks** list does not contain the **min-vendor:css** task, click **Refresh**.

□ 3. In **Task Runner Explorer**, under **Tasks**, right-click **min-vendor:css**, and then click **Run**.

> ℹ **Note**: In Solution Explorer, under **wwwroot**, under **css**, a new css file has been added named **vendor.min.css**.

□ 4. In **Task Runner Explorer**, under **Tasks**, right-click **min-vendor:js**, and then click **Run**.

□ 5. In the **Microsoft Visual Studio** dialog box, click **Yes**.

> ℹ **Note**: In Solution Explorer, under **wwwroot**, under **scripts**, a file named **vendor.min.js** was updated.

Task 3: Style the application by using Bootstrap

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, expand **Views**, expand **Shared**, and then click **_Layout.cshtml**.

□ 2. In the **_Layout.cshtml** code window, locate the following code:

```
<title>@ViewBag.Title</title>
```

□ 3. Place the cursor after the **>** (greater than) sign of the **</title>** tag, press Enter, and then type the following code:

```
<script src="~/scripts/vendor.min.js"></script>
<script src="~/scripts/script.min.js"></script>
<link href="~/css/vendor.min.css" rel="stylesheet" />
<link href="~/css/main.min.css" rel="stylesheet" />
```

□ 4. In the **_Layout.cshtml** code window, locate the following code:

```
<div>
    @RenderBody()
</div>
```

□ 5. Place the cursor before the **<** (less than) sign of the **<div>** tag, press Enter, press the Up arrow key, and then type the following code:

```
        <a class="navbar-brand" href="@Url.Action("Index", "IceCream")">
            <img src="~/images/brand.jpg" class="d-inline-block align-top" alt="">
            Ice Cream of Dreams
        </a>
        <div class="collapse navbar-collapse" id="nav-content">
            <ul class="navbar-nav" id="nav-content">
                <li class="nav-item active">
                    <a class="nav-link" href="@Url.Action("Index", "IceCream")">Hom
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="@Url.Action("Buy", "IceCream")">Buy I
                </li>
            </ul>
        </div>
    </nav>
</div>
<div class="main-title">
        <h1>The Best Ice Cream You Will Taste in Your Life</h1>
        <button type="button" onclick="location.href='@Url.Action("Buy", "IceCream"
</div>
```

□ 6. In the **IceCreamCompany - Microsoft Visual Studio** window, in Solution Explorer, expand **Controllers**, and then click **IceCreamController.cs**.

□ 7. In the **IceCreamController.cs** code window, right-click the following code, and then click **Add View...**.

```
[HttpGet]
public IActionResult Buy()
```

□ 8. In the **Add New Scaffolded Item** dialog box, ensure that the **Razor View** template (not the Empty template) is selected, and click **Add**.

□ 9. In the **Add Razor View** dialog box, ensure that the value in the **View name** textbox is **Buy**.

□ 10. In the **Add Razor View** dialog box, ensure that the **Create as a partial view** check box is cleared and the **Use a layout page** check box is selected, and then click **Add**.

□ 11. In the **Buy.cshtml** code window, place the cursor at the beginning of the document, type the following code, and then press Enter.

```
@model IceCreamCompany.Models.Customer
```

□ 12. In the **Buy.cshtml** code window, select the following code:

```
<h1>Buy</h1>
```

```
<div class="container">
    <h1>Choose Your Flavor</h1>
    <div class="checkout">
        <div class="row justify-content-center intro-row">
            <div class="col-4">Ice Cream Flavors</div>
            <div class="col-2">Buy Bulk(lbs)</div>
            <div class="col-2">Total Amount</div>
            <div class="col-2"></div>
        </div>
        <div class="row justify-content-center">
            <div class="col-4">
                <select class="form-control" id="flavor">
                    <option>Select</option>
                    <option>Vanilla Ice Cream with Caramel Ripple and Almonds</opti
                    <option>Vanilla Ice Cream with Cherry Dark Chocolate Ice Cream<
                    <option>Vanilla Ice Cream with Pistachio</option>
                </select>
            </div>
            <div class="col-2">
                <select class="form-control" id="quantity">
                    <option>1</option>
                    <option>1.5</option>
                    <option>2</option>
                    <option>3</option>
                    <option>4</option>
                </select>
            </div>
            <div class="col-2">
                <div id="totalAmount"></div>
            </div>
            <div class="col-2">
                <div>
                    <img class="imageDisplay" id="iceCreamImage" alt="">
                </div>
            </div>
        </div>
    </div>
</div>
```

☐ 14. In the **Buy.cshtml** code window, place the cursor before the **<** (less then) sign of the last **</div>** tag, press Enter, press the Up arrow key, and then type the following code:

```
        <form method="post" enctype="multipart/form-data" asp-action="Buy">
            <div class="form-group row">
                <label asp-for="FirstName" class="col-sm-4 col-form-label"></label>
                <div class="col-sm-6">
                    <input asp-for="FirstName" type="text" class="form-control" pla
                </div>
            </div>
            <div class="form-group row">
                <label asp-for="LastName" class="col-sm-4 col-form-label"></label>
                <div class="col-sm-6">
                    <input asp-for="LastName" type="text" class="form-control" plac
                </div>
            </div>
            <div class="form-group row">
                <label asp-for="Address" class="col-sm-4 col-form-label"></label>
                <div class="col-sm-6">
                    <input asp-for="Address" type="text" class="form-control" place
                </div>
            </div>
            <div class="form-group row">
                <label asp-for="Email" class="col-sm-4 col-form-label"></label>
                <div class="col-sm-6">
                    <input asp-for="Email" type="email" class="form-control" placeh
                </div>
            </div>
            <div class="form-group row">
                <label asp-for="PhoneNumber" class="col-sm-4 col-form-label"></labe
                <div class="col-sm-6">
                    <input asp-for="PhoneNumber" type="number" class="form-control"
                </div>
            </div>
            <div class="form-group row">
                <div class="col-sm-10">
                    <button id="formButton" type="submit" class="btn btn-outline-pr
                        Make a Purchase
                    </button>
                </div>
            </div>
        </form>
    </div>
</div>
```

Task 4: Run the application

□ 1. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

□ 4. On the **Buy Ice Cream** page, in the **Ice Cream Flavors** list, make a selection.

□ 5. On the **Buy Ice Cream** page, in the **Buy Bulk(lbs)** list, select a value.

□ 6. On the **Buy Ice Cream** page, in the **First Name** box, type your first name.

□ 7. On the **Buy Ice Cream** page, in the **Last Name** box, type your second name.

□ 8. On the **Buy Ice Cream** page, in the **Address** box, type some text.

□ 9. On the **Buy Ice Cream** page, in the **Email** box, type **bill@foo.com**.

□ 10. On the **Buy Ice Cream** page, in the **Phone Number** box, type some numbers and then click **Make a Purchase**.

□ 11. On the **Thanks for your purchase and hope you enjoy the ice cream!** page, on the menu bar, click on the home link at the top of the page, and examine the browser content.

> **Note:** Try changing the width of your browser window. The ice cream images start to stack vertically when there isn't enough space to arrange them horizontally. This adaptation of the layout to suit the display width is an example of Responsive Design.

□ 12. In Microsoft Edge, click **Close**.

□ 13. In the **IceCreamCompany - Microsoft Visual Studio** window, on the **File** menu, click **Exit**.

> ✓ **Results**: In this exercise, you created an ice cream company application, styled using the Bootstrap CSS framework, in which users can view ice cream details, and buy some as well.