**Module 3.2: Laboratory Exercises**

**Exercise 1: Displaying other Information from data.**

#### Task 1: Displaying the product brand and product description

1. In the Start Menu click **Visual Studio Code**
2. In the **Visual Studio Code** click **File** then click **Open Folder.**
3. In the dialog box navigate to **D:\ITLCANGULAR\MOD3.2\Starter\Exercise1\LaptopWebApplication,** then click **Select Folder.**
4. In the **Visual Studio Code Explorer,** expand **src** folder then expand **app** folder then click **app.component.html.**
5. In the **app.component.html,** locate the following code.

**<em class=”float-right”>{{laptop.price}}</em>**

**</h3>**

1. After the located code, add the following code.

**<section></section>**

1. Between the **section** element, add the following code.

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Brand</h4>**

**<h4 class=”card-title”>Description</h4>**

**</div>**

**</div>**

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Specification</h4>**

**</div>**

**</div>**

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Reviews</h4>**

**</div>**

**</div>**

1. In the **app.component.html**, locate the following code.

**<h4 class=”card-title”>Brand</h4>**

1. Below the located code, add the following code.

**<p class=”card-text”>{{laptop.brand}}</p>**

1. In the **app.component.html,** locate the following code.

**<h4 class=”card-title”>Description</h4>**

1. Below the located code, add the following code.

**<p class=”card-text”>{{laptop.description}}</p>**

1. In the **Visual Studio Code** window, click on **File** then click on Save **all.**
2. In the start menu click **Node.js command prompt,** then navigate to **D:\ITLCANGULAR\MOD3.2\Starter\Exercise1\LaptopWebApplication** by entering the command.

**>D:**

**>cd** **\ITLCANGULAR\MOD3.2\Starter\Exercise1\LaptopWebApplication**

1. In the **Node.js command prompt** enter the following command to start the server.

**>ng serve**

1. In the start menu click on Google Chrome, then enter the following address.

**http://localhost:4200**

1. In the Google Chrome window, verify that the web application now displays all the product with its Brand and Description.
2. In the Google Chrome window, click on **Close.**

#### Task 2: Displaying the product specifications.

1. In the **app.component.html,** locate the following code.

**<h4 class=”card-title”>Specifications</h4>**

1. Below the located code, add the following code.

**<ul class=”list-group”>**

**<li class=”list-group-item”>Processor: {{laptop.specification.processor}}</li>**

**<li class=”list-group-item”>Operating System: {{laptop.specification.os}}</li>**

**<li class=”list-group-item”>Memory: {{laptop.specification.memory}}</li>**

**<li class=”list-group-item”>Graphics: {{laptop.specification.graphics}}</li>**

**<li class=”list-group-item”>Display Size: {{laptop.specification.displaysize}}</li>**

**<li class=”list-group-item”>Storage: {{laptop.specification.storage}}</li>**

**</ul>**

1. In the **Visual Studio code** window, click on **File** then click on **Saveall.**
2. In the Google Chrome window, verify that the web application now displays all the product with its Brand, Description and Specifications.
3. In the Google Chrome window, click on **Close.**

#### Task 3: Displaying the product reviews.

1. In the **app.component.html,** locate the following code.

**<h4>Reviews</h4>**

1. Below the located code, add the following code.

**<blockquote \*ngFor=”let review of laptop.reviews”>**

**</blockquote>**

1. Between the blockquote element add the following code.

<**strong>Rating: {{review.stars}}</strong>**

**<cite>Author:{{review.author}}</cite>**

**<cite>Date:{{review.createdOn}}</cite>**

**<p>**

**{{review.body}}**

**</p>**

1. In the **Visual Studio code** window, click on **File** then click on Save **all.**
2. In the Google Chrome window, verify that the web application now displays all the product with its Brand, Description, Specifications and Reviews.
3. In the Google Chrome window, click on **Close.**
4. In the Visual Studio Code window, click on **Close.**
5. In the **Node.js** command prompt, click on **Close.**

**Exercise 2: Creating the method for Reviews.**

#### Task 1: Creating the form for Reviews

1. In the Start Menu click **Visual Studio Code**
2. In the **Visual Studio Code** click **File** then click **Open Folder.**
3. In the dialog box navigate to **D:\ITLCANGULAR\MOD3.2\Starter\Exercise2\LaptopWebApplication,** then click **Select Folder.**
4. In the **Visual Studio Code Explorer,** expand **src** folder then expand **app** folder then click **app.component.html.**
5. In the **app.component.html** locate the following code.

**<p>**

**{{review.body}}**

**</p>**

**</blockquote>**

1. Below the located code add the following code.

**<form name=”reviewForm”>**

**</form>**

1. Between the form element, add the following code.

**<blockquote>**

**<strong>Rating:{{review.stars}}</strong>**

**<cite>Author:{{review.author}}</cite>**

**<cite>Date:{{review.createdOn}}</cite>**

**<p>{{review.body}}**

**</p>**

**</blockquote>**

1. After the code that you have just added, add the following code.

**<div class=”form-group”>**

**</div>**

1. Inside the **div** element add the following code.

**<select name=”stars” class=”form-control” required>**

**<option value=”1”>1 star</option>**

**<option value=”2”>2 stars</option>**

**<option value=”3”>3 stars</option>**

**<option value=”4”>4 stars</option>**

**<option value=”5”>5 stars</option>**

**</select>**

1. After the code that you just added add the following code.

**<textarea name=”body” placeholder=”write a short review of the product…” class=”form-control” required>**

**</textarea>**

**<label>by:</label>**

**<input name=”author” type=”email” required class=”form-control”/>**

**<input type=”submit” value=”Submit Review” class=”form-control”/>**

1. In the **Visual Studio Code** window, click on **File** then click on Save **all.**
2. In the start menu click **Node.js command prompt,** then navigate to **D:\ ITLCANGULAR\MOD3.2\Starter\Exercise2\LaptopWebApplication** by entering the command.

**>D:**

**>cd** **\ITLCANGULAR\MOD3.2\Starter\Exercise2\LaptopWebApplication**

1. In the **Node.js command prompt** enter the following command to start the server.

**>ng serve**

1. In the start menu click on Google Chrome, then enter the following address.

**http://localhost:4200**

1. In the Google Chrome window, under the reviews of the first Laptop, Click the Rating dropdown then click **“5 stars”.**
2. In the reviews text area type **“Test Review”.**
3. In the author text box type, **johndoe@mail.com** then click **Submit Review.**
4. The review can’t be submitted at this moment.
5. In the Google Chrome window, click on **Close.**

#### Task 2: Creating the Method for Reviews.

1. In the **Visual Studio Explorer**, expand **src** folderthen expand **app** folder thendouble click **app.component.ts.**
2. In the **app.component.ts,** add the following codes before the laptops declaration.

**review={};**

**addReview=function(laptop){**

**this.review.createdOn=Date.now();**

**this.laptop.reviews.push(this.review);**

**this.review={};**

**}**

1. In the **Visual Studio Explorer**, navigate to **src/app** folder thendouble click **app.module.ts**
2. Inside the **app.module.ts** locate the following code.

**import { NgModule } from ‘@angular/core’;**

1. Below the located code, add the following code.

**import { FormsModule } from ‘@angular/forms’;**

1. Inside the **app.module.ts** locate the following code.

**imports: [**

**BrowserModule,**

1. Below the located code, add the following code.

**FormsModule**

1. In the **Visual Studio code** window, click on **File** then click on Save **all.**

#### Task 3: Modifying the app.component.html to use the Review Method

1. In the **Visual Studio Explorer,** double click the **app.component.html**
2. In the **app.component.html** locate the following code.

**<form name=”reviewForm”>**

1. Replace the located code with the following code.

**<form name=”reviewForm” (ngSubmit)=”addReview(laptop)” #form=”ngForm”>**

1. In the **app.component.html** locate the following code.

**<select name=”stars” class=”form-control”>**

1. Replace the located code with the following code.

**<select name=”stars” class=”form-control” [(ngModel)]=”review.stars”>**

1. In the **app.component.html** locate the following code.

<**textarea name=”body” placeholder=”Write a short review of the product..” class=”form-control”>**

1. Replace the located code with the following code.

**<textarea name=”body” placeholder=”Write a short review of the product..” class=”form-control” [(ngModel)]=”review.body”>**

1. In the **app.component.html** locate the following code.

**<input name=”author” type=”email” class=”form-control”/>**

1. Replace the located code with the following code.

**<input name=”author” type=”email” class=”form-control” [(ngModel)]=”review.author” />**

1. In the **Visual Studio Code** window, click on **File** then click on Save **all.**
2. In the Solution Explorer, right click **index.html** then click **View in Browser(Google Chrome).**
3. In the Google Chrome window, under the reviews of the first Laptop, Click the Rating dropdown then click **“5 stars”.**
4. In the reviews text area type **“Test Review”.**
5. In the author text box type, **johndoe@mail.com** then click **Submit Review.**
6. Verify that the Test Review is added to the reviews of the first laptop
7. In the Google Chrome window, click on **Close.**
8. In the Visual Studio window, click on **Close.**

*Note: The data for review will disappear when the page is refreshed, this is because the data was not sent to the server to be persisted.*

**Exercise 3: Adding Features to a component**

#### Task 1: Creating the navigation panels

1. In the start menu click the **Visual Studio Code** then
2. In the **Visual Studio Code** click on file then click open folder
3. In the dialog box navigate to **D:\ ITLCANGULAR\MOD3.2\Starter\Exercise3\LaptopWebApplication** then click **select folder**.
4. In the **Visual Studio Explorer** double click **app.component.html.**
5. In the **app.component.html ,** locate the following code.

**<section>**

1. After the located code, add the following code.

**<ul class=”nav nav-pills”>**

**</ul>**

1. Inside the UL element add the following codes.

**<li class=”nav-item”><a class=”nav-link” href>Description</a></li>**

**<li class=”nav-item”><a class=”nav-link” href>Specification</a></li>**

**<li class=”nav-item”><a class=”nav-link” href>Reviews</a></li>**

1. In the start menu click **Node.js command prompt** then navigate to **D:\ ITLCANGULAR\MOD3.2\Starter\Exercise3\LaptopWebApplication** by entering the following commands.

**>D:**

**>cd ITLCANGULAR\MOD3.2\Starter\Exercise3\LaptopWebApplication**

1. In the **Node.js command prompt** enter the following command.

**>ng serve**

1. In the start menu click on Google Chrome then browse the following address.

**http://localhost:4200**

1. In the Google Chrome window, verify that it now displays 3 buttons for Description, Specification and Reviews.
2. In the Google Chrome Window click close.

#### Task 2: Creating the method for panels.

1. In the **Visual Studio Explorer** expand **src** then expand **app** then double click **app.component.ts.**
2. In the **app.component.ts** code window, add the following code to create the default selected panel.

**tab=1;**

**lapindex=0;**

1. After the code that you just added, add the following code to create a function that will handle the selection of panels.

**selectTab=function(laptopindex,setTab){**

**this.tab=setTab;**

**this.lapindex=laptopindex;**

**};**

1. After the code that you just added, add the following code to create a function that will check if the panel is selected.

**isSelected=function(laptopindex,checkTab){**

**return (this.tab===checkTab)&& (this.lapindex===laptopindex);**

**};**

1. In the **Visual Studio code,** click on **File** then click **Save All.**

#### Task 3: Modifying the Webpage.

1. In the **Visual Studio Code Explorer** double click **app.component.html**
2. In the **app.component.html** locate the following code.

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Brand</h4>**

1. Replace the located code with the following code.

**<div class=”card” \*ngIf=”isSelected(i,1)”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Brand</h4>**

1. In the **app.component.html** locate the following code.

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Specification</h4>**

1. Replace the located code with the following code.

**<div class=”card” \*ngIf=”isSelected(i,2)”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Specification</h4>**

1. In the **app.component.html** locate the following code.

**<div class=”card”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Reviews</h4>**

1. Replace the located code with the following code.

**<div class=”card” \*ngIf=”isSelected(i,3)”>**

**<div class=”card-body”>**

**<h4 class=”card-title”>Reviews</h4>**

1. In the **app.component.html** locate the following code.

**<li class=”nav-item”><a class=”nav-link” href>Description</a></li>**

**<li class=”nav-item”><a class=”nav-link” href>Specification</a></li>**

**<li class=”nav-item”><a class=”nav-link” href>Reviews</a></li>**

1. Replace the located code with the following code.

**<li class=”nav-item”><a class=”nav-link” [ngClass]=”{active:isSelected(i,1)}” (click)=”selectTab(i,1)”>Description</a></li>**

**<li class=”nav-item”><a class=”nav-link” [ngClass]=”{active:isSelected(i,2)}” (click)=”selectTab(i,2)”>Specification</a></li>**

**<li class=”nav-item”><a class=”nav-link” [ngClass]=”{active:isSelected(i,3)}” (click)=”selectTab(i,3)”>Reviews</a></li>**

1. In the **Visual Studio Code,** click on **File** then click **Save All.**
2. In the start menu click **Node.js command prompt** then navigate to **D:\ ITLCANGULAR\MOD3.2\Starter\Exercise3\LaptopWebApplication** by entering the following commands.

**>D:**

**>cd ITLCANGULAR\MOD3.2\Starter\Exercise3\LaptopWebApplication**

1. In the **Node.js command prompt** enter the following command.

**>ng serve**

1. In the start menu click on Google Chrome then browse the following address.

**http://localhost:4200**

1. In the Google Chrome window, verify that it now displays 3 buttons for Description, Specification and Reviews.
2. In the web page click the Specification Button
3. In the web page click the Reviews Button
4. In the web page click the Description Button
5. Verify that the panel switches accordingly.
6. In the Google Chrome Window click close.

**Exercise 4: Creating Child Components**

#### Task 1: Creating the Child Component for LaptopsList

1. In the start menu click on **Visual Studio Code.**
2. In the **Visual Studio Code,**  click **File** then click **Open Folder.**
3. In the dialog box to **D:\ITLCANGULAR\MOD3.2\Starter\Exercise4\LaptopWebApplication,** then click **Select Folder.**
4. In the **Visual Studio Code Explorer,** expand **Src** folder then expand **app** folder.
5. Right click the **app** folder then click **New folder.**
6. Name the **New Folder** to **laptops.**
7. Right click the **laptops** folder then click **New File.**
8. Name the **New File** as **laptoplist.component.ts**
9. Right click the **laptops** folder then click **New File.**
10. Name the **New File** as **laptoplist.component.html**
11. In the **laptops** folder, click the **laptoplist.component.ts** file.
12. In the **laptoplist.component.ts** file add the following codes

**import { Component } from ‘@angular/core’**

**@Component({**

**selector: ’laptop-list’,**

**templateUrl:’laptoplist.component.html’**

**})**

**export class LaptopListComponent {**

**}**

1. In the **Visual Studio Code** window, click file then click **Save All.**

#### Task 2: Refactoring the app.component.html

1. In the **Visual Studio Code Explorer,** click the **app.component.html**
2. In the **app.component.html** locate the following code.

**<div class=”container”> -contents- </div>**

1. **Highlight** then **cut** the –**contents-** of the div element that you have located.
2. In the **Visual Studio Code Explorer,** under **laptops folder** click the **laptoplist.component.html**
3. In the **laptoplist.component.html,** **paste** the code that you **cut** from **app.component.html**
4. In the **app.component.html,** locate the following code.

**<div class=”container”>**

1. Below the located code add the following code.

**<laptop-list ><laptop-list>**

1. In the **Visual Studio Code Explorer**, click **File** then click **Save All.**

#### Task 3: Passing Data to the Child Component.

1. In the **Visual Studio Code Explorer** click **laptoplist.component.ts**
2. In the **laptoplist.component.ts,** locate the following code.

**import { Component } from ‘@angular/core’**

1. Replace the located code with the following code.

**import { Component, Input } from ‘@angular/core’**

1. In the **laptoplist.component.ts,** locate the following code.

**export class LaptopListComponent {**

1. Below the located code add the following code.

**@Input() laptops:any;**

1. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**
2. In the **Visual Studio Code** **Explorer** click **app.component.html**
3. In the **app.component.html** locate the following codes.

**<laptop-list></laptop-list>**

1. Replace the located code with the following.

**<laptop-list [laptops]=”laptops”></laptop-list>**

1. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**

#### Task 4: Creating the Child Component for Reviews

1. In the **Visual Studio Code Explorer,** expand **src** folder then expand **app** folder.
2. Right click the **app** folder then click **New folder.**
3. Name the **New Folder** to **laptopreviews.**
4. Right click the **laptopreviews** folder then click **New File.**
5. Name the **New File** as **reviews.component.ts**
6. Right click the **laptopreviews** folder then click **New File.**
7. Name the **New File** as **reviews.component.html**
8. In the **laptopreviews** folder, click the **reviews.component.ts** file.
9. In the **reviews.component.ts** file add the following codes

**import { Component } from ‘@angular/core’**

**@Component({**

**selector: ’laptop-reviews,**

**templateUrl:’reviews.component.html’**

**})**

**export class LaptopReviewsComponent {**

**}**

1. In the **Visual Studio Code** window, click file then click **Save All.**

#### Task 5: Refactoring the laptoplist.component.html

1. In the **Visual Studio Code Explorer,** click the **laptoplist.component.html**
2. In the **laptoplist.component.html** locate the following codes.

**<h4>Reviews</h4>**

**<blockqoute> … </blockqoute>**

**<form> … </form>**

1. **Highlight** then **cut** the **located codes**.
2. In the **Visual Studio Code Explorer,** under **laptopreviews folder** click the **reviews.component.html**
3. In the **reviews.component.html,** **paste** the code that you **cut** from **laptoplist.component.html**
4. In the **laptoplist.component.html,** add the following code to the place where you **cut** the **located codes**.

**<laptop-reviews></laptop-reviews>**

1. In the **Visual Studio Code Explorer**, click **File** then click **Save All.**

#### Task 6: Passing Data to the Child Component.

1. In the **Visual Studio Code Explorer** click **reviews.component.ts**
2. In the **reviews.component.ts,** locate the following code.

**import { Component } from ‘@angular/core’**

1. Replace the located code with the following code.

**import { Component, Input } from ‘@angular/core’**

1. In the **reviews.component.ts,** locate the following code.

**export class LaptopReviewsComponent {**

1. Below the located code add the following code.

**@Input() laptopdetails:any;**

1. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**
2. In the **Visual Studio Code** **Explorer** click **laptoplist.component.html**
3. In the **laptoplist.component.html** locate the following codes.

**<laptop-reviews ></laptop-reviews >**

1. Replace the located code with the following.

**<laptop-reviews [laptopdetails]=”laptop”></laptop-reviews>**

1. In the **Visual Studio Code** click **reviews.component.html.**
2. In the **reviews.component.html,** locate the following code.

**<blockquote \*ngFor="let review of laptop.reviews">**

1. Replace the located code with the following code.

**<blockquote \*ngFor="let review of laptopdetails.reviews">**

1. In the **reviews.component.html,** locate the following code.

**<form #form="ngForm" name="reviewForm" (ngSubmit)="addReview(laptop)">**

1. Replace the located code with the following code.

**<form #form="ngForm" name="reviewForm" (ngSubmit)="addReview(laptopdetails)">**

1. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**

#### Task 7: Moving the methods to the child components

1. In the **Visual Studio Code,** click the **app.component.ts**
2. In the **app.component.ts** locate and **cut** the following codes.

**tab=1;**

**lapindex=0;**

**selectTab=function(laptopindex,setTab){**

**this.tab=setTab;**

**this.lapindex=laptopindex;**

**};**

**isSelected=function(laptopindex,checkTab){**

**return (this.tab===checkTab)&& (this.lapindex===laptopindex);**

**};**

1. In the **Visual Studio Code,** click the  **laptoplist.component.ts**
2. In the **laptoplist.component.ts** locate the following codes.

**@Input() laptops:any;**

1. Below the located code **paste** the code that you’ve **cut**.
2. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**
3. In the **Visual Studio Code,** click the **app.component.ts**
4. In the **app.component.ts** locate and **cut** the following codes.

**review={};**

**addReview=function(laptop){**

**this.review.createadOn=Date.now();**

**laptop.reviews.push(this.review);**

**this.review={};**

**};**

1. In the **Visual Studio Code,** click the  **reviews.component.ts**
2. In the **reviews.component.ts** locate the following codes.

**@Input() laptopdetails:any;**

1. Below the located code paste the code that you’ve **cut**.
2. In the **Visual Studio Code** window, click on **File** then click **Save** **all.**

#### Task 8: Register the components into the Module.

1. In the **Visual Studio Code** Explorer click the **app.module.ts** file.
2. In the **app.module.ts** file locate the list of imports.
3. In the list of imports add the following codes.

**import { LaptopListComponent } from './laptops/laptoplist.component';**

**import { LaptopReviewsComponent } from './laptopreviews/reviews.component';**

1. In the **app.module.ts** file locate the following code.

**declarations:[**

**AppComponent**

1. Replace the located code with the following code.

**declarations:[**

**AppComponent,**

**LaptopListComponent,**

**LaptopReviewsComponent**

1. In the **Visual Studio Code** window**,** click **File** then click **Save All.**
2. In the start menu click **Node.js command prompt.**
3. In the **Node.js command prompt** enter the following command.

**>D:**

**>cd ITLCANGULAR\MOD3.2\Starter\Exercise4\LaptopWebApplication**

1. In the **Node.js command prompt** enter the following command.

**>ng serve**

1. In the start menu open Google Chrome then browse the following address.

**http://localhost:4200**

1. Verify that the LaptopWebApplication stills behaves the same, but now we are using child components.