**Module 6: Laboratory Exercises**

**Exercise 1: Fixing the UI using Angular Pipes**

#### Task 1: Using Angular Pipes to format the currency

1. In the start menu click on **Visual Studio Code,** then click **File** then click **Open Folder.**
2. In the dialog box navigate to **D:\ ITLCANGULAR\MOD6\Starter\Exercise1\LaptopWebApplication,** then click **select folder.**
3. In the **Visual Studio Explorer** expand the **src** folder then expand the **app** folder then expand the **laptops** folder then click **laptoplist.component.html.**
4. In the **laptoplist.component.html** locate the following code.

**{{laptop.price}}**

1. Replace the located code with the following code.

**{{laptop.price | currency}}**

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

**>D:**

**> cd ITLCANGULAR/MOD6/Starter/Exercise1/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 address

**http://localhost:4200**

1. In the Google Chrome window verify that price for the laptop is now formatted with the dollar sign.
2. In the **laptoplist.component.html,** locate the following code.

**{{laptop.price | currency}}**

1. Replace the located code with the following code.

**{{laptop.price | currency: “Php ”}}**

1. In the **Visual Studio Code** window, click on **File** then click on **Save** **all.**
2. Switch to Google Chrome then verify that price for the product is now formatted with the Php sign.

#### Task 2: Using Angular Pipes to format the date

1. In the **Visual Studio Explorer** expand **laptopreviews** folder then click **reviews.component.html.**
2. In the **reviews.component.html,** locate the first blockquote element.
3. Inside the blockquote element locate the following code.

**{{review.createdOn}}**

1. Replace the located code with the following code.

**{{review.createdOn | date}}**

1. In the **reviews.component.html,** locate the second blockquote element.
2. Inside the blockquote element locate the following code.

**{{review.createdOn}}**

1. Replace the located code with the following code.

**{{review.createdOn | date}}**

1. In the **Visual Studio Code** window, click on **File** then click on **Save** **all.**
2. Switch to Google Chrome, in the webpage click on **Reviews,** verify that the date for the reviews is now in date format.
3. In the Google Chrome window click close.
4. In the Node.js command prompt click close
5. In the Visual Studio code window click close

**Exercise 2: Adding Validations**

#### Task 1: Disabling the Post request

1. In the start menu click Visual Studio code, then click on **File** then click **Open Folder**
2. In the dialog box navigate to **D:\ITLCANGULAR\MOD6\Starter\Exercise2\LaptopWebApplication,** then click **Select Folder.**
3. In the **Visual Studio Solution Explorer** expand **src** folderthen expand **app** folder then expand **laptopreviews** folder then click **reviews.component.ts.**
4. In the **reviews.component.ts** inside the **addReview** function locate the following code then comment it out (this is only temporary).

**//this.Reviews.postReviews(this.newReviews).subscribe();**

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

**>D:**

**> cd ITLCANGULAR/MOD6/Starter/Exercise2/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 address

**http://localhost:4200**

1. In the webpage click on **Reviews,** in the Review forms, click the **Submit Review** button.
2. Verify that when the **Submit Review** button is clicked even without data the form is still submitted.

#### Task 2: Adding Validations

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

**#form=”ngForm**”

1. Replace the located code with the following code.

**#revform=”ngForm”**

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

**(ngSubmit)=”addReview(laptopdetails)”**

1. Replace the located code with the following code.

**(ngSubmit)=”revform.form.valid && addReview(laptopdetails);revform.reset()”**

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

**<select name="stars"**

1. After the located code add the following code.

**#stars="ngModel"**

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

**<textarea name="body"**

1. After the located code add the following code.

**#body="ngModel"**

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

**<input name="author"**

1. After the located code add the following code.

**#author ="ngModel" pattern="^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$"**

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

#### Task 3: Adding Validation messages

1. In the **reviews.component.html** locate the **select** element.
2. Before the **select** element add the following code.

**<div class="alert alert-danger" \*ngIf="stars.invalid && (stars.dirty || stars.touched)" >**

**<div \*ngIf="stars.errors.required">Please choose your rating!</div>**

**</div>**

1. In the **reviews.component.html** locate the **textarea** element.
2. Before the **textarea** element add the following code.

**<div class="alert alert-danger" \*ngIf="body.invalid && (body.dirty || body.touched)">**

**<div \*ngIf="body.errors.required">Body is required!</div>**

**</div>**

1. In the **reviews.component.html** locate the **input** element for author.
2. Before the **input** element for author add the following code.

**<div class="alert alert-danger" \*ngIf="author.invalid && (author.dirty || author.touched)">**

**<div \*ngIf="!author.errors.required">Email is required!</div>**

**<div \*ngIf ="!author.errors">Email is invalid</div>**

**</div>**

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

**<input type=”submit”**

1. After the located code add the following codes.

**[disabled]=”!revform.valid”**

1. In the **Visual Studio Code** window, click on **File** then click on **Save** **all.**
2. Switch to Google Chrome window, then refresh the web page.
3. In the webpage click on **Reviews,** in the Review forms, verify that the submit button is disabled.
4. In the Ratings drop down select **5 stars.**
5. In the body text area type **“This is a nice laptop”**, delete the text
6. Verify that the validation error appears when the textarea is empty
7. In the author text box type “johndoe”
8. Verify that the validation error appears when the email is not in the correct format.
9. In the body textarea type **“This is a nice laptop”**
10. In the author textbox type **“johndoe@mail.com”**
11. Verify that the submit button is enabled.

#### Task 4: Enabling the Post request

1. In the **Visual Studio Solution Explorer** expand **src** folderthen expand **app** folder then expand **laptopreviews** folder then click **reviews.component.ts.**
2. In the **reviews.component.ts** inside the **addReview** function locate the following code

**//this.Reviews.postReviews(this.newReviews).subscribe();**

1. Uncomment the located code.
2. In the **Visual Studio Code** window, click on **File** then click on Save **all.**
3. In the Google Chrome window, In the webpage click on **Reviews,** in the Review forms, click the **Submit Review** button.
4. Verify that when the **Submit Review** button is clicked even without data the form will not be submitted.
5. In the Google chrome window click **Close.**
6. In the Visual Studio Code window click **Close.**
7. In the Node.js command prompt click **Close.**