GlobalLogic®
A Hitachi Group Company

Angular HandsOn Assessment

Mobato

- Title of the Project - Mobato
- Complexity - Medium
- Target Band - Band x / Band 1 / Band 2
- Downloadable Starter Code link: <<Link>>
- Skills: HTML,CSS,Typescript,Angular.
- Time taken to complete: 3hrs
- IDE: Visual Studio Code

Technology	Topics
HTML, CSS, Typescript, Angular	Components, Data Binding, Directives, Services, Routing, Forms

Mobato

Mobato is an efficient Angular application designed for mobile device management. It includes an AddMobileComponent for adding new devices using Reactive Forms, ViewMobileComponent to display mobile details, and a HomeComponent welcoming users with an option to add new devices.

Backend Running:

Before you can run the frontend, the backend must be running. Your task is to write a script that will start the backend JAR file and then start the frontend Angular application.
The script should do the following:
1.Use the java -jar command to start the backend JAR file. The JAR file is located in a specific directory, and you should replace <path-to-jar> with the actual path to the JAR file and <your-jar-file> with the name of your JAR file. The command should look like this: java -jar <path-to-jar>/<your-jar-file>.jar.
2.After the backend has started, use the ng serve command to start the frontend Angular application. The Angular application is located in a specific directory, and you should replace <path-to-angular-app> with the actual path to the Angular application. The command should look like this: cd <path-to-angular-app> && ng serve.

Front End Implementation

Component 1: Add Mobile

1.You are tasked with creating an Angular component named AddMobileComponent.ts. This component will be responsible for adding a new mobile device to an inventory system. The component should include a form that allows the user to enter the details of a new mobile device. The form should include fields for the brand, model, price, color, storage, and a description of the mobile device.

2.The form should use Angular's Reactive Forms module for form handling and validation. The form should be initialized in the ngOnInit lifecycle hook and should include appropriate validators for each field.

3.For example, if the brand field is left empty, the form should display an error message saying that the brand is required. If the brand field is filled with a single character, the form should display an error message saying that the brand must be at least 2 characters long.

4.The same applies to the other fields, with the price and storage fields also checking that the entered value is not less than 0, and the description field checking that the entered text is not less than 10 characters long.

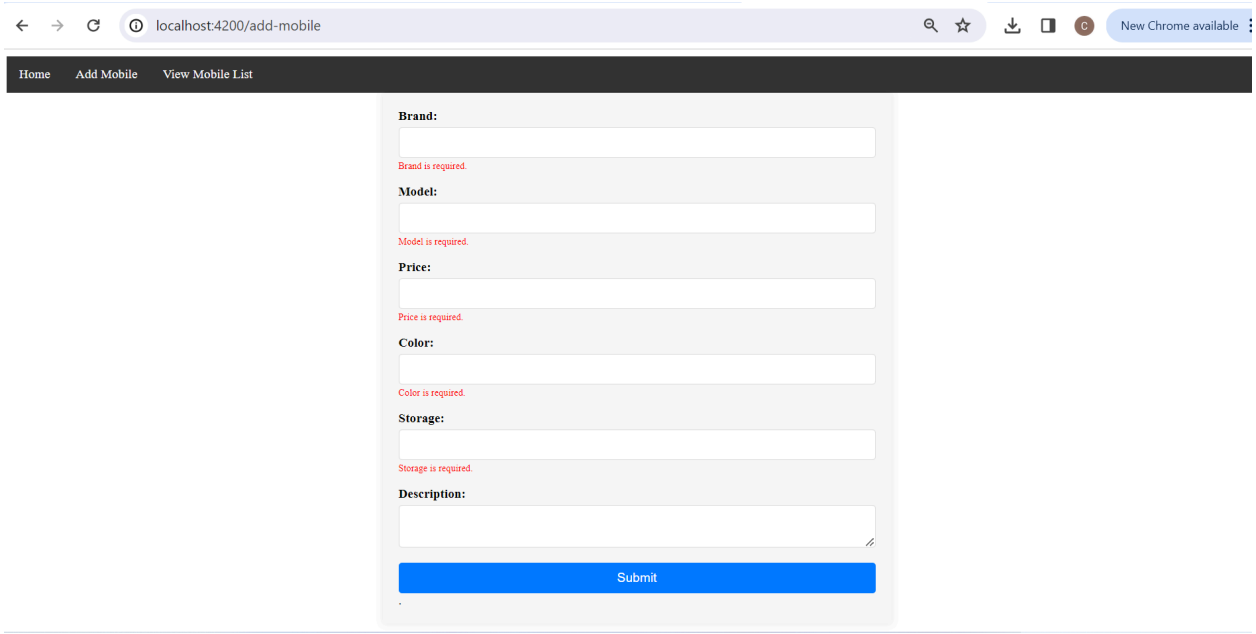
5.Upon form submission, the submit method should be called. This method should clear any previous success or error messages and then call the addMobile method from an AppService service, passing in the form's current values.

6.The addMobile method is expected to return an Observable. If the Observable emits a successful response, the successMessage property should be updated with a message from the response. If the Observable emits an error, the errorMessage property should be updated with an error message from the error response.

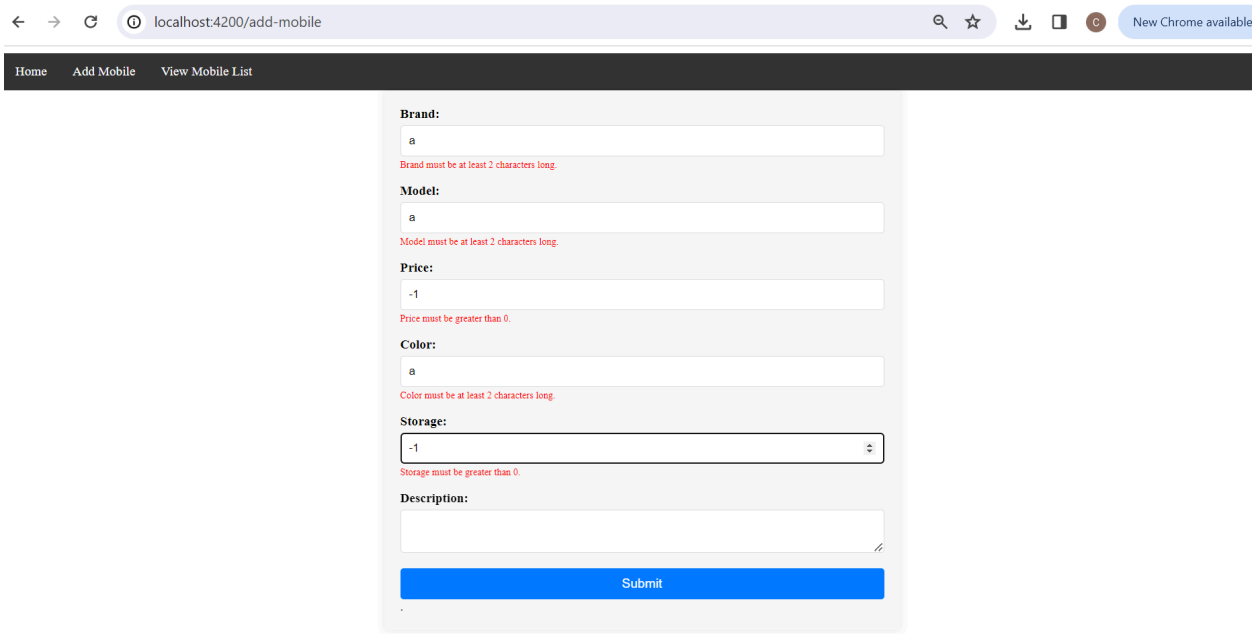
7.Your task is to implement the AddMobileComponent as described above. The TypeScript code for the component

8.You are given a form in Angular that is used to add a new mobile device to an inventory. The form includes fields for the brand, model, price, color, storage, and a description of the mobile device. The form uses Angular's Reactive Forms module for form handling and validation.

Screenshot 1:



Screenshot 2:

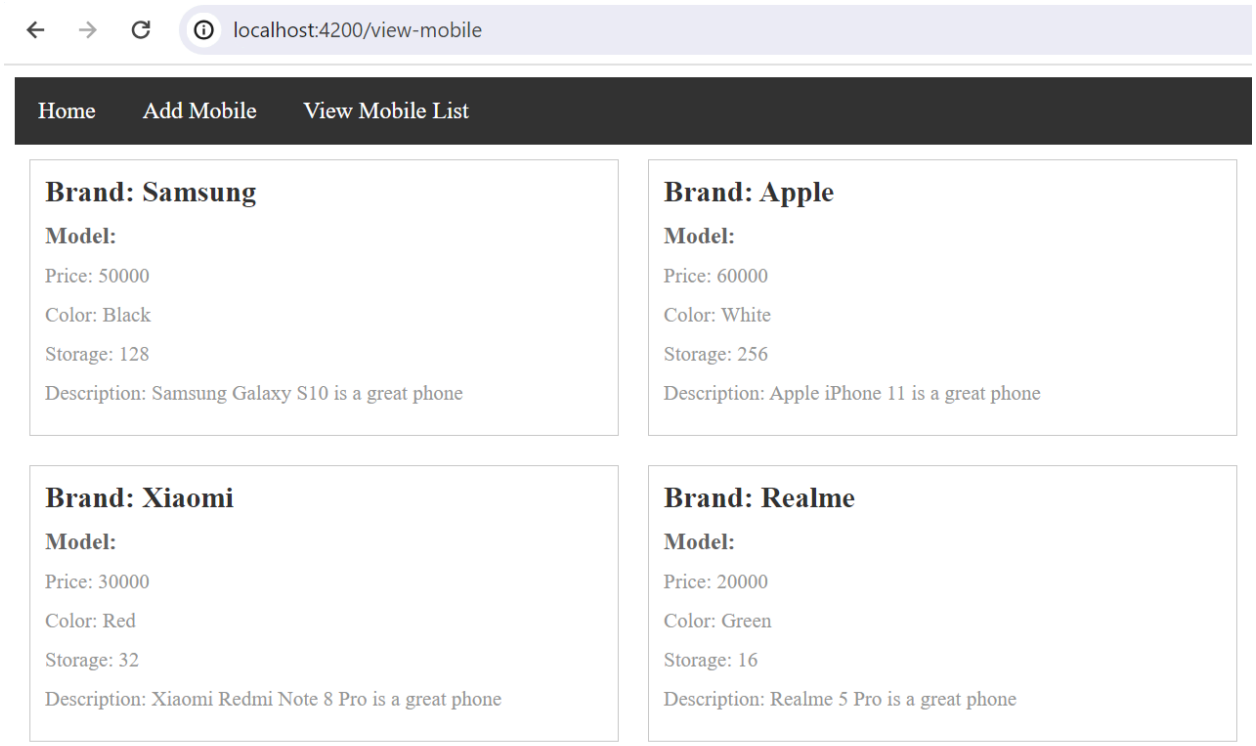


Component 2: View Mobile

1.Implement an Angular component ViewMobileComponent.ts. This component should display the details of all mobiles. It should use the viewMobile method from AppService to get the list of all mobiles.

2.Write the HTML template for ViewMobileComponent.html . The template should display the details of all mobiles in a card format

Screenshot 3:

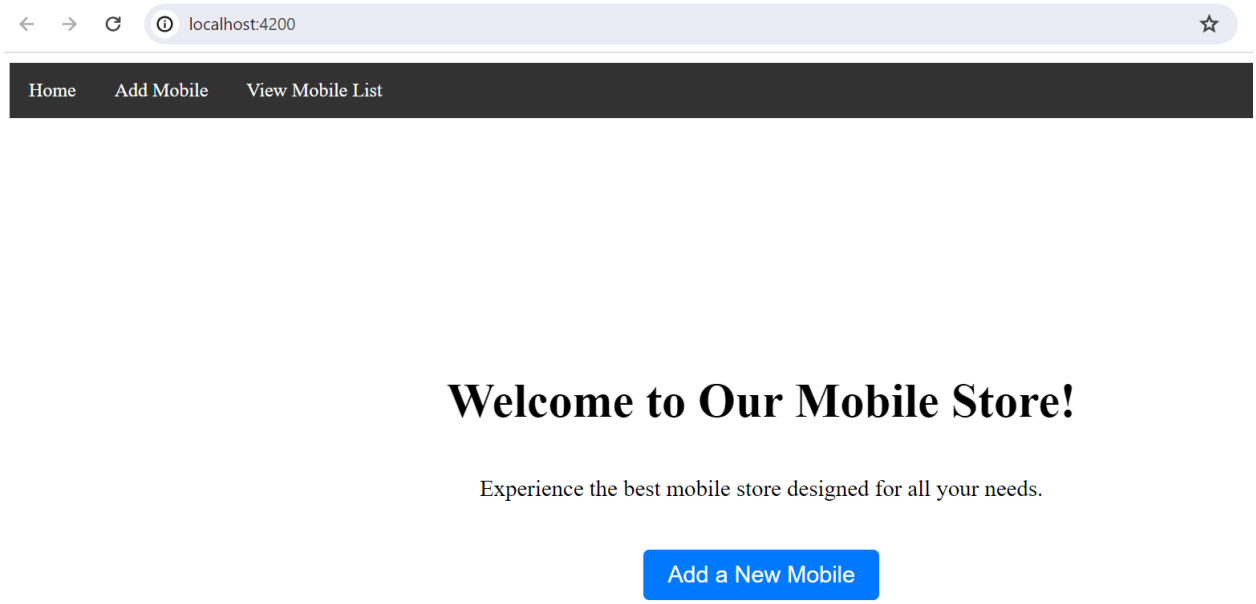


Component 3: Home Component

1.Implement an Angular component HomeComponent.html. This component should display a welcome message to the user.

2.The HomeComponent should also have a button to add a new mobile. When this button is clicked, the application should navigate to the AddMobileComponent

Screenshot 4:



Component 4: App Component

1.Implement an Angular component AppComponent. This component should not have any specific logic, as it serves as a container for the application.

2.Write the HTML template for AppComponent. The template should include a navigation bar with links to the HomeComponent, AddMobileComponent, and ViewMobileComponent. Use the provided app.component.html as a reference. The template should also include a <router-outlet> element where the content of the other components will be displayed.

Component 5: App Service

1.Implement an Angular service AppService. This service should have two methods: addMobile and viewMobile.
2.The addMobile method should take a Mobile object as a parameter and send a POST request to "http://localhost:8080/Mobato/add-mobile" with the mobile data. It should return an Observable.

3.The viewMobile method should send a GET request to "http://localhost:8080/Mobato/get-all-mobile" and return an Observable of an array of Mobile objects.