Ionic Components and TypeScript-10

In this assignment, you will expand on your previous lab by adding UI components from the Ionic library and also writing some TypeScript.

1. Ensure you commit, push, and submit your commit

2. Open Visual Studio code and open your app from the last lab.

3. Let’s first deal with our remote repository. Go over to GitHub and create a new repository for this lab.

4. Next we will run a few commands in VS Code to disconnect our previous remote and add our new report. Open the terminal in VS Code and run the following command to disconnect your current remote repository:

   ```
   git remote remove origin
   ```

5. Run the following command (be sure to use your new URL) to add your new repository:

   ```
   git remote add origin https://github.com/<your new repository URL>
   ```

6. Do a push and check GitHub to ensure your files pushed to the new repository.

7. Let’s add some UI components. You can view all the available UI components here: https://ionicframework.com/docs/components. First, let’s add a button. From the file explorer on the left side, open the **src** folder, then the **app** folder, then the **tab2** folder and open **tab2.page.html**. Insert the following code between the ion-content tags:

   ```
   <ion-button shape="round" fill="outline">My First Button</ion-button>
   ```

8. Next, we will add a content card to our screen. Insert the following code after your button:

   ```
   <ion-card>
     <ion-card-header>
      <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
      <ion-card-title>Card Title</ion-card-title>
     </ion-card-header>

     <ion-card-content>
      Card content.
     </ion-card-content>
    </ion-card>
   ```

9. Let’s add one more element, then we will do some TypeScript coding. Ionic has many out of the box icons you can you use. You can find them here: . Let’s add a few icons to our screen, insert the following code after your content card:

   ```
   <ion-icon name="volume-high"></ion-icon>
   <ion-icon name="volume-low"></ion-icon>
   <ion-icon name="volume-mute"></ion-icon>
   <ion-icon name="volume-off"></ion-icon>
   ```

10. Okay, now let’s make our button do something. Start by updating your button tag to declare a click event by changing it to the following:

    ```
    <ion-button shape="round" fill="outline" (click)="showAlert();">My First Button</ion-button>
    ```

11. Now, let’s go write the function showAlert(). In the **tab2** folder, open the **tab2.page.ts** file (.ts stands for TypeScript). Insert the following code after the constructor:

    ```
      showAlert() {
        alert("Testing the alert pop up.");
      }
    ```

12. Run your app using the following command:

    ```
    ionic serve
    ```

13. After you have verified your UI elements are showing and your button works, kill the app by pressing Ctrl+C in the terminal.

14. Let’s run our app on an Android device (this must be done on a computer with Android Studio). Plug in an Android device, ensure it is setup as a developer with USB debugging on (just like we did in our Android course), and run the following command:

    ```
    ionic cordova run android
    ```