IT 332 Mobile Application Development

Assignment # 2

Last Date: 28-Oct-2019

Task: Create an Ionic application that displays a List (this could be any list e.g movies list, games list, books list, etc) but not a students list. When I click on one of the item, it should open a new page with detail. There should also be a back button at the top. You need to be able to add a new item to that list. You can do that using a different page. Then you would need to add a way to delete any item. Make sure it asks for an alert message before deleting. All this should be done using Angular Services. Once done, take proper screenshots of your application and add them to the readme.md file, and upload to your github account.

Summarizing:

Display a list (hint: *ngFor)

Display an item detail (hint: display detail in new page)

Add Item to the list (hint: use angular forms)

Delete Item from the list

Steps:

1. Create a sample new ionic application.

2. Since we need a new page to display our list, lets create a new page using **ionic g**

3. Now since we need to display the list, we need to create that Array. We will do that in a Service file. So lets create a service. **ionic g** and choose service.

4. add the array to the service and add a function which returns the list. You will find a lot of help from my code from lecture 9 (https://github.com/alamgirqazi/MAD-workbooks/tree/lecture9).

5. Now we display the list in the new page / component we created by calling the function from the service in ngOnInit or ionViewDidEnter(). Before doing this, we need to inject the service in the constructor like
constructor(private mylistService: MylistService) and import it.

6. Now we need to display the detail of a single item to a new full page. For that, we will first need to create a new page using **ionic g**, update the app-routing file.

Using something like

```
{
  path: 'home',
  loadChildren: () => import('./home/home.module').then(m => m.HomePageModule)
},
{
  path: 'studentslist',
  children: [
    {
      path: '',
      loadChildren:
        './studentslist/studentslist.module#StudentslistPageModule'
    },
    {
      path: ':studentid',
      loadChildren: './student/student.module#StudentPageModule'
    }
  ]
}
```

7. After the page is created, we need to either add a **RouterLink** or a function which takes us the the new page.

8. Once we get to the new page, our URL should include the **id** of that item. Something like localhost:8100/studentslist/**3200**
we need to get that ID and match it from our Array in the service. Once the match is done, we can display the detail of that item.

9. Lets add the back button at the top that takes us back.

10. Try to **git add .** , **git commit –m "did task 1"** , **git push** after every step that you do.

11. Add a delete button which will delete an item from the list. Make sure it asks for a prompt. Once deleted, it should take you back to the previous page.

12. We need to add a new student. Lets create a new page. Add a Angular Reactive Form, and save it to the list. You proper Alerts and Validations. Now we move to the list page, the new student should be there.

13. Make sure you create a **readme.md** file. The readme file should have screenshots of all the page of your application. If anyone visits your github link, he would get to know what your application does.

This assignment covers a lot of very important concepts that you need to learn for building applications. I strongly recommend you go through Lecture 8 and 9 and also the link to code. Clone the repository, run it locally and this assignment would be much easier. For Step 12, go through Lecture 6. Again, if you are stuck anywhere, you can email me. Try to write the problem in detail, provide screenshot and its best if you provide me the link of your github repository.

Everything is difficult before it is easy.

Good Luck!