Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Student Full Name.
2. This is a group assignment, each group is expected to make one submission.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Create a zip file which includes all the project folder, any required libraries, and your presentation material.
6. Submission details:
   a. You should submit the assignment through canvas: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

In Class 06 (100 Points)

In this assignment you will get familiar with ListViews/RecyclerViews and making simple API requests. You will develop a contacts application which enable you to list, create, edit and delete contacts. The contacts api details are described in the provided Postman file.

Your application should contain only **one Activity and multiple fragments** to implement the required screens. The wireframes are not provided to leave it up to you to design your apps. All API calls should be done through the OkHttp library which should be integrated in your project.

Part 1 : Contacts List Screen (30 Points)
The requirements are as follows:
1. This is the main launcher screen:
   a. It should retrieve the list of users using the "/contacts" API.
   b. You should create a Contact class to store the contact information.
   c. Each displayed list item should show all the contact information in addition to a Delete button.
2. Clicking on a row item should go to the Details Screen. The Contact List Screen should pass the required information to the Details Screen in order to display the contact details.
3. Clicking the "Delete" button should delete the selected contact and should refresh the List to show the updated list of contacts.
   a. The app should call the "/contact/delete" API after the API is completed you should reload the list of users calling the "/contacts" API and refresh the list.
4. Clicking the "Add" button should go to the New Contact Screen.

Part 2 : New Contact Screen (20 Points)
The requirements are as follows:
1. Clicking the "Cancel" button should dismiss this Screen and return back to the Contacts List Screen.
2. Clicking the "Submit" button, the app should check if all the fields are entered and display an *alert dialog* if any of the entries is missing indicating that the missing field is required.
   a. If all the fields are entered, the app should call the "/contact/create" API. After the API is completed the app should dismiss this Screen and signal the Contacts List Screen to refresh the list of users.
   b. Upon returning to the Contacts List Screen the List should be refreshed and should include the newly created contact. This should be done by reloading the list of users by calling the "/contacts" API and refresh the list.

Part 3 : Contact Details Screen (25 Points)
The requirements are as follows:
1. This screen should display the contact details.
2. Clicking the "Delete" button should perform the following steps:
    a. Call the "/contact/delete" API. After the API is completed this Screen should signal the Contacts List Screen indicating that this contact was deleted successfully.
    b. Dismiss the current Screen, which should show the Contacts List Screen.
    c. The Contacts List Screen should reload the list of users calling the "/contacts" API and refresh the list.
3. Clicking the "Update" button should go to the Edit Contact Screen and should pass the contact object to the Edit Contact Screen.

Part 4 : Edit Contact Screen (25 Points)
The requirements are as follows:
1. This Screen should display the edit form populated with the details of the contact that was passed by the Details Screen.
2. Clicking the "Update" button, the app should check if all the fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
    a. If all the fields are entered, the app should call the "/contact/update" API. After the API is completed the app should dismiss this Screen and signal the pervious Contact Details Screen to refresh the contact details. In addition should signal the Contacts List Screen that the contact information was updated.
    b. Upon returning to the Contact Details Screen the contact details should show the updated contact information.
    c. Upon returning to the Contacts List Screen the List should be refreshed and should include the newly created contact. This should be done by reloading the list of users by calling the "/contacts" API and refresh the list.