Hello everyone, my name is Arya Dewa Wijaya, ok I will answer these five questions:

1.What project have you worked about?

In this project I am working on a sheduler app, which can show the nearest shedule on main page, show other schedule on list page also can delete it with swipe action, show detail schedule, change theme into dark or light mode and show notification of today schedule.

2.Which part is hardest?

For me the most difficult thing was implement daily reminder. Becuae I need to integrated between alarm manager and notification, and I have a problem with set daily reminder every 6 a.m. with calendar API because the notification sometimes didnot work.

3.How is the flow to change the theme to a dark theme?

The flow is that when the user chooses to change the theme to a dark theme, there are options which are provided using a ListPreference, the item of choice is displayed as a string array. The user will select a Dark mode item which has the key pref\_dark\_on and the value ON. When clicked there will be a listener that handles it, after that the value of the preference will be changed to capital and checked with the provided enum class nightmode. Because the preference key it has is pref\_dark\_on which has a value of ON where the appropriate is AppCompatDelegate.MODE\_NIGHT\_YES which has a value of 2, after that the theme will be updated with AppCompatDelegate.setDefaultNightMode() with the parameter entered is the value 2 and the theme will be changed to dark theme

4.How does notification reminder work?

In this project we use BroadcastReceiver, that receive the Today Schedule and passing it to notification. For the notification we use alarmManager set into repeat alarm at 6 a.m. and passing it with pending intent that contain id repeating. To show the data we use inbox style and get the data from broadcast receiver

Then create notification object by using NotificationCompat.Builder and pass the notifcation channel id. We also can add other configuration like icon, title, context, sound, and the pending intent same as created before that contain id repeating but now configure it to open HomeActivity went user click the notification.

Then check if the user's phone is Android version Oreo and above because no need to create a notification channel for older versions.

Then create the notification using the NotifcationManagerCompat, passing in the Notification channel ID and the notification Object

and that's how notifications work

5.Why do we need LiveData? lifecycle

Because LiveData to store data, do it on the addcourseviewmodel class. LiveData notifies the Observer object when the data changes. So it automatically updates the UI. With liveData, we don't need to update the UI every time when the app data changes because the observer does it and with this there are no memory leaks. live data also automatically manages process cycle changes

when making observations. it prevents crashes caused due to destroyed Activity or Fragment.