

25th July 2025 (Friday)

Today's class was dedicated to exploring Firebase Cloud Messaging (FCM), one of the most powerful tools for enabling real-time notifications in Android apps. The mentor began with a theoretical overview of FCM architecture — explaining how messages are sent from a cloud server to a client app using unique device tokens. We learned that push notifications play an important role in improving user engagement, especially for chat apps where users expect instant alerts.

During the first hour, I studied the complete workflow of FCM. The trainer explained the concept of tokens — unique identifiers assigned to each app instance — and how these tokens allow Firebase to send notifications specifically to a user's device. We discussed the types of notifications: display messages (visible notifications) and data messages (background processing). The mentor also highlighted real-world scenarios like how WhatsApp and Messenger handle message alerts even when users are offline.

In the second hour, we moved to practical implementation. We configured Firebase Cloud Messaging by integrating the necessary dependencies and updating the google-services.json file in our Android project. Then, I created a class extending FirebaseMessagingService to override the onMessageReceived() and onNewToken() methods. Using Firebase Console, we manually sent test notifications to verify delivery. When my emulator displayed a notification titled "New Message Received" even though the app was minimized, I felt genuinely excited — it was the first time my app interacted with the cloud outside of the main interface.

Afterward, I customized the notification behavior — adding vibration, LED light, and sound alerts using NotificationCompat.Builder. I also created custom intents so that tapping the notification opened the corresponding chat screen. The mentor reviewed my implementation and appreciated the completeness of the feature.

This session taught me the importance of user re-engagement and made me realize how backend services can make a mobile app feel "alive." I ended the day by documenting the code and flow diagram for notification handling to include in my final project report.