

Android Development with Kotlin(126 hours)

Course Objectives:

- Understand the basics of Android development, Kotlin programming, and the Android Studio environment.
-

- Design and develop user interfaces, implement data storage, and manage network communication for Android applications.
- Deploy and manage Android applications.

Course Outcomes:

- **CO1:** Recall the fundamental concepts of Android development and Kotlin programming. *(Remembering)*
- **CO2:** Explain the Android app components and lifecycle. *(Understanding)*
- **CO3:** Apply user interface design principles to develop Android applications. *(Applying)*
- **CO4:** Analyze data storage and network communication techniques. *(Analyzing)*
- **CO5:** Develop and deploy comprehensive Android applications. *(Creating)*

Course Syllabus:

Module 1: Introduction to Android Development (16 hours)

Theory

- Overview, Android Architecture
- Setting up Android Studio, Kotlin Basics
- Creating First Android Project

Practice

- Experiment 1.1: Set up Android Studio and create a new project.
- Experiment 1.2: Write a Kotlin program to print "Hello, World!".
- Experiment 1.3: Implement basic arithmetic operations in Kotlin.
- Experiment 1.4: Create a simple Android app with a single activity.
- Experiment 1.5: Design a user interface using XML layout.
- Experiment 1.6: Implement basic navigation between activities.
- Experiment 1.7: Demonstrate the use of Kotlin data types and variables.
- Experiment 1.8: Implement a simple calculator app in Android.

Module 2: Android App Components (16 hours)

Theory

- Activities, Fragments, Services
- Broadcast Receivers, Content Providers
- Intents, Activity Lifecycle

Practice

- Experiment 2.1: Implement an activity lifecycle in Android.
 - Experiment 2.2: Create and use fragments in an Android app.
 - Experiment 2.3: Implement a service to run in the background.
 - Experiment 2.4: Use broadcast receivers to receive system events.
 - Experiment 2.5: Implement content providers to share data between apps.
 - Experiment 2.6: Use intents to navigate between activities.
 - Experiment 2.7: Implement pending intents for notifications.
 - Experiment 2.8: Demonstrate the use of shared preferences.
-

Module 3: User Interface Design (16 hours)

Theory

- Layouts: Linear, Relative, Constraint, Frame
- UI Components: TextView, EditText, Button, ImageView, RecyclerView
- Material Design Principles

Practice

- Experiment 3.1: Design a user interface using ConstraintLayout.
- Experiment 3.2: Implement a RecyclerView to display a list of items.
- Experiment 3.3: Create custom views and view groups.
- Experiment 3.4: Implement material design components in Android.
- Experiment 3.5: Use themes and styles to customize the UI.
- Experiment 3.6: Implement user input controls (TextView, EditText, Button).
- Experiment 3.7: Create a form with input validation.
- Experiment 3.8: Implement a navigation drawer for app navigation.

Module 4: Data Storage (16 hours)

Theory

- SharedPreferences, Internal, External Storage
- SQLite Database, Room Persistence Library
- Content Providers

Practice

- Experiment 4.1: Implement SQLite database in an Android app.
- Experiment 4.2: Use Room Persistence Library for data storage.
- Experiment 4.3: Implement content providers for data sharing.
- Experiment 4.4: Use shared preferences for simple data storage.
- Experiment 4.5: Implement internal storage for file management.
- Experiment 4.6: Implement external storage for file management.
- Experiment 4.7: Use LiveData to observe data changes.
- Experiment 4.8: Implement data binding in an Android app.

Module 5: Network Communication (16 hours)

Theory

- HTTP Networking, RESTful APIs
- Retrofit, Volley Libraries
- JSON Parsing, Background Tasks with WorkManager

Practice

- Experiment 5.1: Implement HTTP networking using Retrofit.
- Experiment 5.2: Implement HTTP networking using Volley.
- Experiment 5.3: Parse JSON data from a web API.
- Experiment 5.4: Implement background tasks using WorkManager.

-
- Experiment 5.5: Use OkHttp for network communication.
 - Experiment 5.6: Implement WebSockets for real-time communication.
 - Experiment 5.7: Use Firebase Realtime Database for data synchronization.
 - Experiment 5.8: Implement RESTful API communication.

Module 6: Advanced Topics (18 hours)

Theory

- Firebase Integration: Authentication, Realtime Database
- Google Play Services, Location APIs
- App Distribution, Google Play Store, App Signing

Practice

- Experiment 6.1: Implement Firebase Authentication in an Android app.
- Experiment 6.2: Use Google Maps API for location-based services.
- Experiment 6.3: Implement push notifications using Firebase Cloud Messaging.
- Experiment 6.4: Integrate social media login (Facebook, Google) in an Android app.
- Experiment 6.5: Implement data synchronization with Firebase.
- Experiment 6.6: Use ML Kit for image recognition.
- Experiment 6.7: Implement in-app purchases in an Android app.
- Experiment 6.8: Use Google Play Services for analytics.

Project (28 hours)

- Project 1: Develop a personal finance management app.
- Project 2: Create a location-based reminder app.
- Project 3: Develop a social media integration app.