

ABOUT THE COURSE:

Android Application Development:

Course Objectives	<ul style="list-style-type: none">● Strong Command of Kotlin: Master Kotlin syntax, data types, and control flow to proficiently develop Android applications.● Firebase Integration: Integrate Firebase free services, including Authentication, Realtime Database, Cloud Firestore, Cloud Messaging, and other essential services, to enhance app functionality and user experience.● Complete Authentication Flow: Implement a secure and user-friendly login and registration process using Firebase Authentication, ensuring data integrity and user privacy.● Real-time Data Synchronization: Implement real-time data synchronization using Firebase Realtime Database and Cloud Firestore, enabling dynamic and responsive applications that update in real-time.● Project Development Skills: Develop a complete and functional Android app, incorporating Firebase services for enhanced functionality, while adhering to industry best practices to ensure the creation of robust and scalable applications.
Course Outcomes	<ul style="list-style-type: none">● Set up Android Studio and initiating a new Kotlin project. Master Kotlin syntax, data types, and control flow for effective Android app development. Learn to create and execute a simple Kotlin-based Android application seamlessly.● Work with UI components like layouts, views, and widgets within Android Studio, alongside a comprehensive knowledge of the Android Activity lifecycle and efficient user input handling.● Exhibit expertise in implementing RecyclerView and CardView to showcase dynamic lists of data effectively in your Android applications. Implement the intricacies of crafting custom views and view groups to tailor your app's user interface to specific requirements.

	<ul style="list-style-type: none">● Implement Firebase Authentication to facilitate secure user login and registration processes. Develop a comprehensive user authentication flow using Kotlin and Firebase Authentication, ensuring a seamless and secure experience for your app users throughout the login and registration processes.● Create data modeling and execute CRUD operations on Firebase Realtime Database Implement Firebase Cloud Fire store for document-based NoSQL data management and synchronization with Kotlin.
--	---

Course Duration: 45 Hours

Course Content:

Unit 1: Introduction to Kotlin for Android development

Setting up Android Studio and creating a new Kotlin project - Understanding Kotlin syntax, data types, and control flow -Creating and running a simple Kotlin-based Android app

Unit 2: Working with UI components in Android Studio

Working with UI components in Android Studio, such as layouts, views, and widgets Understanding the Android Activity lifecycle and handling user input Implementing basic UI interactions, such as button clicks and text input Building a simple user interface using XML and Kotlin

Unit 3: Working with Kotlin extensions

Implementing RecyclerView and CardView for displaying lists of data Understanding and implementing custom views and view groups Handling screen orientation changes and other lifecycle events

Unit 4: Understanding Firebase services

Setting up Firebase project and adding Firebase to an Android app Implementing Firebase Authentication for user login and registration Building a complete user authentication flow with Kotlin and Firebase Authentication

Unit 5: Building an Android App with Firebase Integration

Understanding data models and implementing CRUD operations on Firebase Realtime Database Implementing Firebase Cloud Firestore for document-based NoSQL database Implementing Firestore data syncing and querying with Kotlin-

Understanding Firebase Cloud Messaging topics and device-to-device messaging
 Implementing Firebase in-app messaging for displaying targeted messages to app users
 Building a complete, functional Android app with Firebase free services using Kotlin-Implementing Firebase Remote Config for remotely managing app configurations
 Implementing Firebase Crashlytics for crash reporting and error tracking
 Implementing Firebase Cloud Storage for storing and retrieving files in the cloud
 Implementing Firebase Dynamic Links for deep linking and app linking

Test Projects:

Use Cases

TABLE OVERALL COURSE LEARNING OUTCOME ASSESSMENT CRITERIA AND USE CASES		
LEARNING OUTCOME	ASSESSMENT CRITERIA	USE CASES
Set up Android Studio and initiating a new Kotlin project. Master Kotlin syntax, data types, and control flow for effective Android app development.	Assess proficiency in configuring Android Studio for Kotlin development basic Kotlin syntax, including data types, control flow, and function declarations. Evaluate the ability to create, execute, and debug a simple Kotlin-based Android application, including designing user interfaces, implementing functionality, and handling common development tasks.	<p>Use Case 1: Developing a Basic Kotlin Android Application</p> <p>Scenario: Alex, a beginner Android developer, is eager to create his first Kotlin-based Android app but needs assistance in knowing about the essentials.</p> <p>Task: Walk Alex through the creation of a simple Android application using Kotlin. Start by explaining Kotlin syntax, data types, and control flow. Then, guide him in designing a basic user interface, implementing functionality, and running the app on an emulator or physical device.</p> <p>Use Case 2: Kotlin App Development Workflow</p>

		<p>in Android Studio</p> <p>Scenario: Daniel, a proficient developer, seeks to streamline his Kotlin app development process in Android Studio.</p> <p>Task: Help Daniel optimize his workflow by demonstrating advanced techniques for setting up Android Studio, creating a new Kotlin project, and leveraging Kotlin's features efficiently. Guide him in adopting best practices for code organization, debugging, and testing within the IDE to enhance productivity and code quality.</p>
--	--	---

<p>Work with UI components like layouts, views, and widgets within Android Studio, alongside a comprehensive Knowledge of the Android Activity lifecycle and efficient user input handling.</p>	<ul style="list-style-type: none"> • UI Component Proficiency: Evaluate the ability to work with layouts, views, and widgets in Android Studio, demonstrating seamless integration of XML and Kotlin for UI development. • Activity Lifecycle and User Input Handling: Assessing of Android Activity lifecycle and proficiency in efficiently handling user input, including basic interactions like button clicks and text input validation. 	<p>Use Case 1: Developing a Basic User Login Interface</p> <p>Scenario: Rachel is tasked with creating a simple user login screen for a new Android app.</p> <p>Task: Guide Rachel through designing the user interface using XML layouts in Android Studio. Teach her how to incorporate text input fields for username and password, along with a login button. Ensure the Android Activity lifecycle to handle user interactions effectively, such as validating input</p>
---	---	--

		<p>on button clicks. Additionally, help her implement basic UI interactions using Kotlin, such as displaying a toast message upon successful login.</p> <p>Use Case 2: Creating a Weather Forecast Display Interface</p> <p>Scenario: James wants to build a weather forecast app that displays weather information to users.</p> <p>Task: Assist James in developing a simple user interface in Android Studio using both XML and Kotlin. Teach him how to design the layout to show weather details like temperature, humidity, and wind speed. Ensure the Android Activity lifecycle to update the UI dynamically based on user input or data changes. Guide him in implementing basic UI interactions, such as refreshing weather data on button clicks, and handling user input for location selection.</p>
--	--	---

<p>Implement RecyclerView and CardView to showcase Dynamic lists of data effectively in your Android applications. Exhibit the intricacies of crafting custom views and view groups to tailor your app's user interface to specific requirements.</p>	<ul style="list-style-type: none">• RecyclerView and CardView Integration: Assess proficiency in integrating RecyclerView and CardView to display dynamic data lists efficiently.• Custom View Implementation and Lifecycle Handling: Evaluate and implementation of custom views and view groups, alongside adept management of screen orientation changes and lifecycle events.	<p>Use Case 1: Creating a Dynamic Product Catalog Interface</p> <p>Scenario: Anna, a developer, needs to design an app interface to display a catalog of products.</p> <p>Task: Guide Anna in implementing RecyclerView and CardView to showcase the product list dynamically. Teach her to populate data from a backend server or local database efficiently. Additionally, help her to create custom views and view groups for special product categories or promotional sections. Ensure Anna can handle screen orientation changes smoothly to maintain a consistent user experience.</p> <p>Use Case 2: Developing a Custom News Feed Interface</p> <p>Scenario: Mark is developing a news app and wants to create a unique interface to display articles.</p> <p>Task: Assist Mark in implementing RecyclerView and CardView to present news articles in a</p>
---	--	--

		<p>scrollable list. Teach him to customize the appearance of each article card using CardView attributes. Additionally, guide him in creating custom views for special content types like videos or interactive widgets. Ensure how to manage lifecycle events to preserve the user's reading position and maintain UI consistency during screen orientation changes.</p>
--	--	---

<p>Set up Firebase projects and seamlessly integrate Firebase services into your Android applications. Learn to implement Firebase Authentication to facilitate secure user login and registration processes. Develop a comprehensive user authentication flow using Kotlin and Firebase Authentication, ensuring a seamless and secure experience for your app users throughout the login and registration processes.</p>	<ul style="list-style-type: none"> • Firebase Setup and Authentication Integration: Assess proficiency in setting up Firebase, adding it to an Android app, and implementing Firebase Authentication for user login and registration. • Complete User Authentication Flow Implementation: Evaluate the ability to develop a seamless user authentication flow using Kotlin and Firebase Authentication, covering UI design, user input handling, and authentication 	<p>Use Case 1: Creating a Secure User Registration System</p> <p>Scenario: Emily is developing a new Android app that requires user registration and login functionality.</p> <p>Task: Guide Emily through setting up a Firebase project and integrating Firebase Authentication into her app. Teach her how to implement user registration and login screens using Kotlin and Firebase Authentication. Ensure how to handle user input for registration, validate credentials, and manage user authentication states effectively.</p>
--	---	---

	<p>state management.</p>	<p>Use Case 2: Building a Seamless User Authentication Experience</p> <p>Scenario: Alex wants to enhance the user authentication flow in his existing Android app.</p> <p>Task: Assist Alex in enhancing his app by building a complete user authentication flow with Kotlin and Firebase Authentication. Help him design intuitive UI screens for user registration, login, and password recovery. Guide him in implementing Firebase Authentication features such as email/password authentication and social login. Ensure how to manage authentication states and provide a smooth user experience throughout the authentication process.</p>
<p>Exhibit data modeling and execute CRUD operations on Firebase Realtime Database. Implement Firebase Cloud Firestore for document-based NoSQL data management and</p>	<ul style="list-style-type: none"> • Firestore Proficiency: Evaluate the learner's ability to process data models and implement CRUD operations on Firebase Realtime Database and Firestore. Assess 	<p>Use Case 1: Building a Social Media Platform</p> <p>Scenario: Sarah is developing a new social media platform and needs to implement robust data management and messaging features.</p> <p>Task: Guide Sarah in implementing Firebase</p>

<p>synchronization with Kotlin.</p>	<p>proficiency in data synchronization, querying, and implementing document-based NoSQL databases using Kotlin.</p> <ul style="list-style-type: none"> • Firestore Service Integration and Utilization: Assess the learner's capability to integrate various Firebase services into an Android app effectively. Evaluate and implementation of Firebase Cloud Messaging, in-app messaging, Remote Config, Crashlytics, Cloud Storage, Dynamic Links, and their respective functionalities in Kotlin-based Android development. 	<p>Realtime Database and Firestore to manage user profiles, posts, and messaging. Teach her to create data models for users and posts, implement CRUD operations for data management, and integrate Firebase Cloud Messaging for real-time notifications. Additionally, help her utilize Firebase Dynamic Links for sharing posts and Firebase In-App Messaging for targeted user engagement.</p> <p>Use Case 2: Creating a Content Management App</p> <p>Scenario: Alex wants to develop an app for managing and sharing content with his team.</p> <p>Task: Assist Alex in building an Android app using Kotlin and Firebase services. Help him implement Firebase Cloud Firestore for storing documents related to projects, Firebase Cloud Storage for uploading and retrieving files, and Firebase Remote Config for remotely managing app configurations. Additionally, guide him in integrating Firebase</p>
-------------------------------------	--	--

		<p>Crashlytics for error tracking and Firebase Analytics for monitoring app usage.</p> <p>Ensure Alex can create a seamless user experience with efficient data syncing, remote configuration management, and crash reporting.</p>
--	--	--

LIST OF FINAL PROJECTS (10 PROJECTS THAT COMPREHENSIVELY COVER ALL THE LEARNING OUTCOME)	
SL.NO	FINAL PROJECT
1	Podcast Plus: A Redux-Inspired Podcast App with Dynamic Themes for Android
2	A Sleep Tracking App for a Better Night's Rest
3	Snack Squad: A Customizable Snack Ordering and Delivery App
4	Compose Input: A Demonstration of Text Input and Validation with Android Compose
5	An Android Application for Keeping Up with the Latest Headlines
6	Adaptive Mail: A Flexible Email Client App
7	Owl-M: A Material Design Study App
8	Wanderlust: A Personalized Travel Planning and Tracking App
9	Money Matters: A Personal Finance Management App
10	ChatConnect - A Real-Time Chat and Communication App