**SYLLABUS**

THE COMPLETE FLUTTER DEVELOPMENT BOOTCAMP

**INTRODUCTION TO DEVELOPMENT WITH FLUTTER**

- Learn to set up a new Flutter project using Android Studio.
- Understand the Widget tree and learn to use pre-made Flutter Widgets for user interface design.
- Learn to incorporate Image and Text Widgets to create simple user interfaces.
- Learn to incorporate App Icons for iOS and Android.
- Learn how to add and load image assets to Flutter projects.
- Run Flutter apps on iOS Simulator, Android Emulator and physical devices.

**CREATING BEAUTIFUL UI WITH FLUTTER FOR BEGINNERS**

- Use Hot Reload and Hot Restart to quickly refresh the app UI and understand when to use each.
- Learning to use the Pubs Yosemite file to incorporate dependencies, custom assets and fonts.
- An introduction to the Widget build() method.
- Learning to use layout widgets such as Columns, Rows, Containers and Cards.
- Incorporating Material icons using the Icons class.

**BUILDING APPS WITH STATE**

- Understand the difference between Stateful and Stateless Widgets and when they should each be used.
- Understand how callbacks can be used detect user interaction in button widgets.
- Understand the declarative style of UI programming and how Flutter widgets react to state changes.
- Learn to import dart libraries to incorporate additional functionality.
- Learn about how variables, data types and functions work in Dart 2.
- Build flexible layouts using the Flutter Expanded widget.
- Understand the relationship between setState(), State objects and Stateful Widgets.

**LEVERAGING FLUTTER PACKAGES TO SPEED UP DEVELOPMENT**

- Learn to use the Dart package manager to incorporate Flutter compatible packages into your projects.
- Understanding the structure of the pubspec.yaml file.
- Incorporate the audioplayers package to play sound.
- Learn more about functions in Dart and the arrow syntax.
- Learn to refactor widgets and understand Flutter's philosophy of UI as code.
SYLLABUS
THE COMPLETE FLUTTER DEVELOPMENT BOOTCAMP

Learn about how lists and conditionals work in Dart.
Learn about classes and objects in Dart and how it apply to Flutter widgets.
Understand Object Oriented Dart and how to apply the fundamentals of OOP to restructuring a Flutter app.
Learn to use Dart Constructors to create customisable Flutter widgets.
Apply common mobile design patterns to structure Flutter apps.
Learn about structuring and organising Flutter apps.

Structuring Flutter Apps

Creating Beautiful UI with Flutter for Intermediates

- Customise apps with Theme widgets.
- Refactoring widgets by extracting them as separate Widget classes.
- Learn about Dart annotations and modifiers.
- Understand the immutability of Stateless and Stateful Widgets and how the screen is updated with the build() method.
- Create custom Flutter Widgets by combining smaller widgets.
- Learn about the difference between final and const in Dart.
- Learn about maps, enums and the ternary operator in Dart.
- Understand that functions are first class objects in Dart and how functions can be passed around as arguments.
- Learn to build multi-screen Flutter apps by learning about routes and the Navigator widget.
- Understand why flutter favours composition vs. inheritance when customising widgets.

Powering Flutter Apps with Backend Data

- Learn about asynchronous programming in Dart and understand how to use async/await and the Futures API.
- Understand Stateful Widget lifecycle methods.
- Handling exceptions in dart with try/catch and throw.
- Use Dart null aware operators to prevent app crashes.
- Getting location data from both iOS and Android.
- Using the http package to perform networking and get live data from open APIs.
- Understanding how to parse JSON data using the dart:convert library.
- Understand how to pass data to State objects via the Stateful Widget.
- Use the TextField Widget to take user input.
- Understand how to pass data backwards using the Navigator widget.
Learn to use hero animations in Flutter apps.
Understanding how the animation controller works and create custom animations.
Learn to use Dart mixins to extend class functionality.
Incorporate Firebase Cloud Firestore into your Flutter apps.
Implement authentication in your Flutter apps with the Firebase Auth package.
Build a scrolling ListView widget to learn how Flutter creates and destroys reusable elements.
Understand Dart Streams to listen to data changes.
Learn to use the Flutter StreamBuild to turn streams of data into widgets that can be rendered.

Learn to use the Cupertino package and build Flutter apps for iOS with UIKit styled Cupertino widgets.
Use the dart:io library to detect runtime platform and create separate UIs for iOS and Android in the same Flutter app.
Learn to use Dart loops to create recurring widgets.

Learn why we need to manage state across our widget tree.
Learn about declarative vs. imperative programming.
Look at how setState works under the hood.
Learn about prop drilling and lifting state up.
Create a todo list app.
Learn about the BottomSheet widget and the ListViewBuilder.
Understand Flutter app architecture design patterns.
Learn to manage state with the Google recommended Provider package.