



Android

Flutter App Development Course for Android & iOS [70 Hrs]

Instructor: Pondit Team

E-Mail:

Phone:

Overview

কাদের জন্যে কোর্স:

- এই কোর্সটি যে কেউ করতে পারবে যদি তার কম্পিউটার সম্পর্কে ধারণা থাকে।
- যারা আসলেই আইটিতে ক্যারিয়ার গড়তে চায় তাদের জন্য।

কেন করবেন এই কোর্সঃ

- আইটি কোম্পানীতে ভালো বেতনে জব করার সুযোগ আছে।
- অনলাইন মার্কেটপ্লেসে প্রচুর চাহিদা রয়েছে।
- দক্ষ ট্রেইনার দিয়ে কোর্স পরিচালনা করানো হয়।

Audience

- ছাত্র/ছাত্রী
- চাকুরীজীবী

Evaluation

Curriculum

1. Introduction

- 1.1. Introduction
- 1.2. What is Flutter?
- 1.3. Why Flutter?
- 1.4. Analysis of flutter app development scope

2. Setup and Installation (Android & iOS)

- 2.1. Install the Flutter SDK
- 2.2. Install Android Studio
- 2.3. Install the Android Emulator

3. Dart Language

- 3.1. Variables
- 3.2. Data Types
- 3.3. Functions
- 3.4. Arrow Functions
- 3.5. Conditionals
- 3.6. Lists

Prerequisites

কম্পিউটার সম্পর্কে ধারণা থাকতে হবে

Course Information

- Duration : 70 Hours
- Day : শুক্রবার এবং শনিবার
- Time : রাত ৯.০০টা থেকে ১১.০০টা পর্যন্ত
- Reg. Start : Aug 7, 2021
- Reg. End : Nov 3, 2021
- Class Start : Nov 5, 2021

- 3.7. Final vs. Const
- 3.8. Enums
- 3.9. Ternary Operator
- 3.10. Maps
- 3.11. Futures, Async & Await
- 3.12. Exception Handling & Null Aware Operators
- 3.13. Loops
- 3.14. Static Modifier
- 3.15. Mixins
- 4. Flutter Apps UI Design
 - 4.1. Container Widgets
 - 4.2. Column & Row Widgets for Layout
 - 4.3. Custom Fonts
 - 4.4. Material Icons
 - 4.5. Flutter Card Widgets
 - 4.6. ListTile Widgets
 - 4.7. New Flutter Project from Scratch
 - 4.8. Hot Reload
 - 4.9. Material Design
 - 4.10. Assets in Flutter
- 5. Flutter State
 - 5.1. Expanded Widget to Create Flexible Layouts
 - 5.2. Interaction with Widget
 - 5.3. Flutter Image Widget
 - 5.4. Stateful vs. Stateless Widgets
- 6. OOP
 - 6.1. OOP - Classes and Objects
 - 6.2. OOP - Abstraction in Action
 - 6.3. OOP - Class Constructors
 - 6.4. OOP - Class Constructors
 - 6.5. OOP - Encapsulation in Action
 - 6.6. OOP - Inheritance in Action
 - 6.7. OOP - Polymorphism in Action
- 7. Flutter and Dart Packages
 - 7.1. Flutter & Dart Packages
 - 7.2. Play Music Across Platforms
 - 7.3. Refactoring App
- 8. Source Code Management - Live Project
 - 8.1. Building a Score Keeper
 - 8.2. Displaying the Questions
 - 8.3. Checking User Answers
 - 8.4. Creating a Question Class
 - 8.5. Adding Back the Score Keeper
- 9. BMI Calculator - Live Project

- 9.1. Flutter Themes
- 9.2. Refactor Flutter Widgets
- 9.3. Custom Flutter Widgets
- 9.4. GestureDetector Widget
- 9.5. Flutter Slider Widget
- 9.6. Customizing Widgets
- 9.7. Composition vs. Inheritance
- 9.8. Multi-Screen Apps Using Routes and Navigation
- 10. Weather - Live Project (Using REST API)
 - 10.1. Location Data From Across Platforms
 - 10.2. Stateful Widget Lifecycle
 - 10.3. Application Programming Interfaces (APIs)
 - 10.4. Networking with the HTTP Package
 - 10.5. JSON Parsing and Dynamic Types
 - 10.6. Spinner Widgets
 - 10.7. Passing Data to a State Object
 - 10.8. TextField Widget for Input
 - 10.9. Navigation Stack
- 11. Messenger Live Project with Firebase
 - 11.1. Flutter Animations
 - 11.2. Flutter Animations Packages
 - 11.3. Android Firebase Project Setup
 - 11.4. iOS Firebase Project Setup
 - 11.5. Firebase Authentication
 - 11.6. Saving Data into Cloud Firestore
 - 11.7. Data Retrieve using Streams
 - 11.8. Flutter ListView
- 12. Note Book - Live Project with Provider
 - 12.1. BottomSheet Widget
 - 12.2. What is the State and How do we Manage it?
 - 12.3. Callbacks in Dart
 - 12.4. Provider Package
 - 12.5. ChangeNotifier to Manage State
- 13. Architecture Patterns
 - 13.1. REST API
 - 13.2. MVC
 - 13.3. MVBM
- 14. Local Persistence
 - 14.1. Shared Preferences
 - 14.2. SQLite Database
- 15. Notification
 - 15.1. Background Notification using Firebase
 - 15.2. Foreground Notification using Flutter Package