



Flutter advanced (ขั้นสูง)



ปี 2020 นี้ถ้าจะเริ่มพัฒนาแอปแบบ Cross Platform (รองรับทั้ง Android / iOS) ตัวเลือกที่น่าสนใจ คงเป็นใครไปไม่ได้นอกจาก "Flutter framework" สูตรร้อนแรงจาก google ที่เป็นกระแสความนิยมในหมู่พัฒนาอย่างรวดเร็ว ด้วยความเรียบง่าย ความเร็ว ประสิทธิภาพของแอปพลิเคชันที่ได้ใกล้เคียงกับ Native (การพัฒนาด้วยภาษาเครื่อง) โดยใช้การพัฒนาด้วยภาษาเพียงภาษาเดียวนั่นก็คือ "ภาษา Dart" ที่มีรูปแบบภาษาคัดล้ายคลึงกับภาษา C และ Java ผสมผสานกับ JavaScript อีกนิดหน่อย ตัวภาษานี้ว่ามีความคลาสสิก และความทันสมัยอยู่ในตัว

สำหรับหลักสูตร Flutter นั้นมีรายละเอียด และสิ่งที่ต้องเรียนรู้ค่อนข้างเยอะ หลากหลายมากทีเดียว ดังนั้นเพื่อให้เนื้อหาไม่เยอะเกินไปซึ่งจะทำให้ผู้เริ่มต้นเรียนรู้ตามไม่ทัน และไม่เข้าใจอย่างถ่องแท้ ทางสถาบัน ไอทีจีเนียสจึงมีการแบ่งเนื้อหาออกเป็น 3 ระดับด้วยกัน (3 หลักสูตรต่อเนื่องกัน) ประกอบด้วย

- Flutter beginner (สำหรับผู้เริ่มต้น)
- Flutter intermediate (ขั้นกลาง)
- Flutter advanced (ขั้นสูง)

สำหรับเนื้อหาใน **Flutter Advanced (ขั้นสูง)** นี้เป็นการต่อยอดจาก flutter intermediate ขั้นกลาง เพื่อให้ผู้เรียนและผู้เข้าอบรมสามารถประยุกต์ใช้งาน flutter ได้อย่างครบถ้วน เนื้อหาจะลึกในหัวข้อที่สามารถนำไปใช้ในงานจริง อาทิ การออกแบบด้วย Material Advanced, การออกแบบแอปส่วนของ iOS ด้วย Cupertino, การทำแอปให้รองรับหลายภาษา , การใช้งานฐานข้อมูลภายในแอปด้วย SQLite , การอัปโหลดไฟล์ภาพผ่าน API, การทำงานกับ Google Firebase, ทำระบบ Chat แบบเรียลไทม์, การอัปโหลดรูปภาพด้วย firebase, พัฒนาระบบแจ้งเตือน , เขียนระบบด้วยโครงสร้างแบบ MVC, จัดการ State ด้วย Bloc, การบันทึกเสียงใน flutter, การเขียนระบบเชื่อมต่อกับ Youtube API, การติดต่อกับอุปกรณ์และเซ็นเซอร์ต่างๆ บนมือถือ และเนื้อหาอื่นๆ อีกมากมาย

เมื่อผู้เข้าอบรมจบหลักสูตรนี้ไปจะมั่นใจได้ว่า ได้ความรู้ ความเข้าใจที่ครบถ้วน ในการนำไปต่อยอดพัฒนาผลงานของตนเองได้อย่างมีประสิทธิภาพต่อไป



วัตถุประสงค์:

- การออกแบบด้วย Material Advanced Components
- การออกแบบแอปส่วนของ iOS ด้วย Cupertino (Building a Cupertino app with Flutter for iOS)
- การทำแอปให้รองรับหลายภาษา (Localization and Multi Language in flutter)
- การใช้ฐานข้อมูลภายในแอปด้วย SQLite (Using SQLite in Flutter)
- การอัปโหลดไฟล์ภาพผ่าน API (Upload Images To REST API)
- การทำงานกับ Google Firebase (Firestore CRUD in Flutter)
- ทำระบบ Chat แบบเรียลไทม์ (Realtime chat with firebase in flutter)
- การอัปโหลดรูปภาพด้วย firebase (Firebase Image Upload)
- พัฒนาระบบแจ้งเตือน (Push Notifications in Flutter using Firebase)
- เขียนระบบด้วยโครงสร้างแบบ MVC (Flutter MVC design pattern)
- จัดการ State ด้วย Bloc (State management with Bloc)
- การบันทึกเสียงใน flutter (Flutter Audio Recorder)
- การเล่นวิดีโอ (Flutter Video Player)
- การเขียนระบบเชื่อมต่อกับ Youtube API (Flutter YouTube API and Video Player)
- การติดต่อกับอุปกรณ์และเซ็นเซอร์ต่างๆ บนมือถือ (Flutter with Device API)
- การเตรียมความพร้อมแอปและอัปโหลดขึ้น Store (Build and published app to store)

กลุ่มเป้าหมาย:

- นักเรียนนักศึกษา
- ครู อาจารย์ วิทยากรที่สนใจ
- นักวิชาการ นักไอที หรือผู้ดูแลระบบ
- ตลอดจนผู้สนใจทั่วไปในการพัฒนาแอปพลิเคชันแบบ Cross Platform ด้วย Flutter



ความรู้พื้นฐาน:

- พื้นฐานการเขียน flutter มาบ้างเล็กน้อย หรือผ่านหลักสูตร flutter for intermediate ชั้นกลางมาแล้ว
- มีพื้นฐานภาษา Dart มาบ้าง หากไม่มีจะมีการปรับพื้นฐานให้ก่อนอบรม
- ไม่จำเป็นต้องเขียนภาษา Swift หรือ Java มาก่อน
- มีพื้นฐานการใช้งาน Mobile app ทั้ง android และ iOS มาบ้าง

ระยะเวลาในการอบรม:

- 30 ชั่วโมง (5 วัน)

ราคาคอร์สอบรม:

- 9,500 บาท

วิทยากรผู้สอน:

- อาจารย์สามิตร โกยม และทีมงาน

คอร์สที่ควรอบรมก่อนหน้า:

- Flutter intermediate (ชั้นกลาง)

คอร์สต่อเนื่องที่แนะนำ:

- ไม่มี



เนื้อหาการอบรม:

Module 1: การออกแบบด้วย Material Advanced Components

- ติดตั้ง flutter และกำหนดสภาพแวดล้อม (Set up your Flutter environment)
- เพิ่มส่วนของ backdrop เมนู (Add the backdrop menu)
- เพิ่มการทำรูปทรงต่างๆ (Add a shape)
- เพิ่มการทำ motion (Add motion)
- เพิ่มเมนูส่วน back layer (Add a menu on the back layer)
- เพิ่มไอคอนแสดงแบรนด์ของแอป (Add a branded icon)

Module 2: การออกแบบแอปส่วนของ iOS ด้วย Cupertino (Building a Cupertino app for iOS)

- ติดตั้ง flutter และกำหนดสภาพแวดล้อม (Set up your Flutter environment)
- สร้างและกำหนดค่าเริ่มต้นสำหรับ Cupertino (Create the initial Cupertino app)
- กำหนดโครงสร้างของแอปเป็น 3 แท็บ (Create structure for a 3-tab app)
- เพิ่มส่วนการจัดการ State (Add state management)
- ใช้ list view แสดงรายการ (List products for sale)
- ทำส่วนการค้นหาข้อมูลใน List (Add product search)
- ปรับแต่งส่วนของข้อมูลลูกค้า (Add customer info)
- เพิ่มส่วนการแสดงวันเวลา (Add date picker)
- เพิ่มส่วนการเลือกข้อมูลใน List (Add selected items for purchase)

Module 3: การทำแอปให้รองรับหลายภาษา (Localization and Multi Language in flutter)

- การเตรียมโปรเจกต์ flutter สำหรับรองรับหลายภาษา (Prepare Your Project)
- เพิ่ม package จัดการหลายภาษาลงในไฟล์ pubspec.yaml (Update pubspec.yaml)
- การสร้าง json ไฟล์แยกภาษาที่ต้องการ (Create the language files)
- สร้าง Delegate Class สำหรับจัดการเลือกภาษา (Create Delegate Class)
- จัดการแสดงผลภาษาในส่วนของ UI (Translating text in the UI)
- บันทึกข้อมูลภาษาที่เลือกไว้ลงในเครื่องด้วยการใช้ provider และ shared_preferences (changing App language manually)
- ปรับแต่งการแสดงผลให้เหมาะสมในงานจริง (Custom for your app)



Module 4: การใช้งานฐานข้อมูลภายในแอปด้วย SQLite (Using SQLite in Flutter)

- การใช้ SQLite ใน flutter เบื้องต้น ติดตั้ง plugin SQFLITE (SQLite in Flutter: SQFLITE plugin)
- สร้างส่วนของ model (Implementing Model Class)
- สร้างส่วนของ helper สำหรับทำเรียกดู เพิ่ม ลบ แก้ไข (Creating DatabaseHelper Class: CRUD Operation)
- เชื่อมต่อฐานข้อมูลกับส่วนแสดงผล (Connect database to UI)
- ปรับแต่งการแสดงผลข้อมูลจาก SQLite ในส่วนการแสดงผล (Custom to display data to UI)

Module 5: การอัปโหลดไฟล์ภาพผ่าน API (Upload Images To REST API)

- สร้างโปรเจกต์สำหรับใช้ทดสอบการอัปโหลดรูป (Prepare and create project)
- เพิ่ม package สำหรับการจัดการ network และจัดการไฟล์ใน pubspec.yaml (Install the dependencies in pubspec.yaml)
- กำหนด layout สำหรับเลือกไฟล์ภาพ (Create layout for add image)
- เรียกใช้งาน image_picker (Adding the Image Picking Feature)
- เขียน Rest API ส่วน backend ด้วย Laravel สำหรับอัปโหลดไฟล์ภาพ (Backend Setup To Handle The Image Upload Process)
- เขียน Logic ส่วนการอัปโหลดไฟล์รูปใน flutter (Implementing The Image Upload Feature)

Module 6: การทำงานกับ Google Firebase (Firestore CRUD in Flutter)

- สร้างโปรเจกต์ flutter (Creating a New Firebase Project)
- ติดตั้ง firebase ลงใน project android (Android Firebase Project Setup)
- ติดตั้ง firebase ลงใน project iOS (iOS Firebase Project Setup)
- Firebase Flutter Packages Setup (iOS & Android)
- Registering Users with Firebase using FirebaseAuth
- Authenticating Users with FirebaseAuth
- Showing a Spinner while the User Waits
- Saving Data into Cloud Firestore
- Listening for Data from Firebase using Streams



- Update Data into Cloud Firestore
- Delete Data into Cloud Firestore
- Turning Streams into Widgets Using the StreamBuilder
- Cloud Firestore Authorisation and Security Rules

Module 7: ทำระบบ Chat แบบเรียลไทม์ (Realtime chat with firebase in flutter)

- Creating an Authentication Form
- Connecting Auth Form & Auth Screen
- Implementing Authentication
- Storing Extra User Data
- Finishing Authentication & Logout
- Adding Firebase Security Rules
- Listening to Chat Messages
- Sending & Ordering Chat Messages
- Creating a Basic Chat Bubble Widget
- Improving the Chat Bubbles
- Testing on Multiple Devices
- Displaying User Names

Module 8: การอัปโหลดรูปภาพด้วย firebase (Firebase Image Upload)

- Adding an Image Picker
- Preparing File Upload & Validating Images
- Uploading Images
- Improving Image Upload
- Running on a Real iOS Device
- Displaying Images



Module 9: พัฒนาระบบแจ้งเตือน (Push Notifications in Flutter using Firebase)

- Creating a Firebase project
- Integrating the firebase_messaging package
- Android-specific configurations
- iOS-specific configuration
- Initializing the packages in Dart-Code
- Sending Notifications with the Firebase console

Module 10: เขียนระบบด้วยโครงสร้างแบบ MVC (Flutter MVC design pattern)

- What is the MVC Design Pattern?
- The MVC Approach
- Install package for mvc
- Create project with mvc structure
- The Controller
- The Model
- The View

Module 11: จัดการ State ด้วย Bloc (State management with Bloc)

- What is BLoC pattern?
- Concepts for Bloc Events, States, Transition
- Implementing the BLoC pattern
- Creating Events
- Creating BLoC
- Creating the UI
- Running the app



Module 12: การบันทึกเสียงใน flutter (Flutter Audio Recorder)

- Install project and add package flutter_sound
- Config on iOS
- Subscriptions
- Default uri path
- Creating instance.
- Starting recorder with listener.
- Stop recorder
- Start player
- Stop player
- Pause player
- Resume player
- Seek player
- Setting volume

Module 13: การเล่นวิดีโอ (Flutter Video Player)

- Setup flutter project
- Add package video_player in pubspec.yaml
- Implement for video player
- Custom to display video in view
- Test on real device



Module 14: การเขียนระบบเชื่อมต่อกับ Youtube API (Flutter YouTube API and Video Player)

- Create flutter project and add "youtube_player_flutter" dependencies
- Config to iOS in Info.plist
- Create project and enable api in google developer console
- Get Youtube Data API Key
- Create Model in flutter
- Create Video modal
- Create api service
- Display video in listview
- Play video in screen UI

Module 15: การติดต่อกับอุปกรณ์และเซ็นเซอร์ต่าง ๆ บนมือถือ (Flutter with Device API)

- WebView
- Bluetooth
- Calendar
- File Picker
- Battery
- NFC Reader

Module 16: การเตรียมความพร้อมแอปและอัปโหลดขึ้น Store (Build and published app to store)

- Preparing the Code
- Preparing the App Configuration
- Preparing Third-Party Services
- Adding Icons & Splash Screens
- Publishing Android Apps
- Publishing iOS Apps