TIE IT Consulting Services - Syllabus for IIMSTC Interns

This syllabus is designed by TIEITCS to equip interns with the necessary skills to develop Flutter applications, integrate them with backends, and incorporate basic machine learning functionalities. It culminates in a final project where interns apply their learning.

Module 1: Introduction to Flutter and Dart (10 hours)

- **1.1 Introduction to Flutter (2 hours)**
  - What is Flutter?
  - Advantages of Flutter
  - Flutter architecture
  - Setting up the development environment
  - Creating a simple "Hello World" app
- **1.2 Dart Fundamentals (8 hours)**
  - Variables, data types, and operators
  - Control flow statements (if-else, loops)
  - Functions and methods
  - Classes and objects
  - Collections (lists, maps, sets)
  - Asynchronous programming (Futures, async/await)
  - Exception handling

Deliverables:

- Simple Dart programs demonstrating language concepts
- A basic Flutter app displaying text and images

Module 2: Building User Interfaces with Flutter (15 hours)

- **2.1 Widgets: The Building Blocks of Flutter UIs (5 hours)**
  - Stateless and Stateful widgets
  - Basic widgets (Text, Image, Container, Row, Column)
  - Layout widgets (ListView, GridView)
  - Interactive widgets (Buttons, TextField, Form)
- **2.2 Material Design and Cupertino Style (4 hours)**
  - Implementing Material Design guidelines
  - Using Cupertino widgets for iOS style
  - Theming and styling
- **2.3 Navigation and Routing (6 hours)**
  - Navigating between screens
  - Passing data between screens
  - Different navigation patterns (e.g., tab-based, drawer)

Deliverables:

- A multi-screen Flutter app with different layouts and interactive elements

- Implementation of Material Design or Cupertino style in the app

Module 3: Backend Integration (15 hours)

- **3.1 Introduction to APIs and RESTful Services (3 hours)**
  - What are APIs?
  - Understanding RESTful principles
  - HTTP methods (GET, POST, PUT, DELETE)
  - JSON data format
- **3.2 Making HTTP Requests in Flutter (6 hours)**
  - Using the http package
  - Sending requests and handling responses
  - Parsing JSON data
  - Error handling
- **3.3 Working with a Backend Framework (6 hours)**
  - Introduction to a backend framework (e.g., Node.js with Express, Flask)
  - Setting up a simple API endpoint
  - Connecting the Flutter app to the backend

Deliverables:

- A Flutter app that fetches data from a public API (e.g., news API, weather API)
- A simple backend API with at least one endpoint
- Integration of the Flutter app with the backend API

Module 4: Introduction to Machine Learning in Flutter (10 hours)

- **4.1 Machine Learning Concepts (4 hours)**
  - Supervised vs. unsupervised learning
  - Common machine learning algorithms (e.g., linear regression, classification)
  - Model training and evaluation
- **4.2 Using Pre-trained Models in Flutter (6 hours)**
  - Integrating TensorFlow Lite models
  - Using a pre-trained model for image classification or object detection
  - Handling model inputs and outputs

Deliverables:

- A Flutter app that uses a pre-trained machine learning model (e.g., image classification)

Module 5: Final Project (15 hours)

- **5.1 Project Planning and Design (3 hours)**
  - Brainstorming project ideas
  - Defining project scope and features
  - Designing the app's user interface
- **5.2 Project Development (10 hours)**
  - Implementing the app's functionality
  - Integrating with a backend (optional)

    - Incorporating machine learning (optional)
- **5.3 Project Presentation (2 hours)**
    - Presenting the final project to the team
    - Demonstrating the app's features
    - Explaining the technical implementation

Deliverables:

- A fully functional Flutter app that incorporates the concepts learned throughout the training
- Project documentation (e.g., README file, design specifications)
- Project presentation

Tools and Technologies:

- Flutter SDK
- Dart programming language
- Visual Studio Code or Android Studio
- Git for version control
- Firebase (optional)
- TensorFlow Lite (optional)
- Backend framework (e.g., Node.js, Flask) (optional)