BCAE0403: MOBILE APPLICATION DEVELOPMENT II

Objective: *To introduce students to cross-platform mobile application development using Flutter and Dart, enabling them to build modern, responsive, and feature-rich applications for Android and iOS platforms.*

Credits:03 ***L–T–P–J:3–0–0–0***

Module No.	Content	Teaching Hours
I	**Introduction to Mobile Applications -** Understanding the Need for Mobile Applications, Evolution of Mobile Technology, Examples of Popular Mobile Apps in Various Domains (Social Media, Finance, Health, Education, Entertainment), Characteristics and Features of Mobile Applications, Introduction to App Stores and App Deployment **Mobile Platforms and Development Approaches -** Overview of Popular Mobile Platforms: Android, iOS, Introduction to Native vs Cross-Platform Development, Basic Concepts of Hybrid Apps, Overview of Development Tools and IDEs (Android Studio, VS Code) **Getting Started with Flutter -** Introduction to Flutter and its Use in Modern App Development, Key Features and Advantages of Flutter, Flutter vs Other Frameworks (Basic Comparison with React Native and Native Development), Understanding Flutter Architecture: Dart Language, Widgets, Engine Basics, **Understanding the Dart Programming Language** - Introduction to Dart, Installing Dart SDK and setting up environment, Using DartPad for coding practice, Dart syntax basics, Writing comments and documentation, Conditional statements and operators, Creating and using functions, Object-Oriented Programming (OOP) concepts in Dart. **Flutter App Development Environment** - Installing Flutter SDK and setup, Setting up IDE (VS Code/Android Studio) for Flutter, Running app on emulator and physical devices, Understanding project structure (pubspec.yaml, lib folder), Introduction to hot reload and debugging techniques. **Building UI with Flutter Widgets -** Understanding Widget Tree and Lifecycle, Stateless vs Stateful widgets, Common widgets: Text, Image, Container, Layout widgets: Row, Column, Stack, Align, Buttons: ElevatedButton, IconButton, MaterialApp and Scaffold, Customizing AppBar, Body, FloatingActionButton, Working with assets, images, fonts, and themes.	19
II	**Navigation and Routing -** Understanding App Structure and Navigation, Navigation using Navigator.push(), pushNamed(), and pop(), Passing Data Between Screens, Named Routing and Route Arguments, Best Practices in Route Management **State management in Flutter –** Stateless vs Stateful widgets, Local state using setState(), Advanced techniques: Provider (simplified), Riverpod (basics and use cases), Overview of BLoC pattern, Introduction to GetX for state and route management. **App Performance Optimization -** Efficient Code Structure for Reusability and Scalability, Hot Reload vs Hot Restart, Reducing App Size by Managing Dependencies. **Version Management and Deployment -** Understanding Semantic Versioning: version: in pubspec.yaml, Commands: flutter build apk, flutter build appbundle, Code Signing with Keytool and Jarsigner**, Overview of iOS App Deployment -** Xcode Setup, Provisioning Profiles, and Certificates, Creating Release-Ready Flutter App. **Tools and Industry Practices** - Using **Git** and GitHub for Collaboration, API Integration using http package, Firebase (Firestore, Realtime Database, Cloud Storage, Firebase Authentication). **Testing Flutter Applications** - Debugging with print() and Debug Console, Unit testing with test package, Widget and Integration Testing, Performance profiling using DevTools, Crash reporting and analytics with Firebase Crashlytics and Analytics.	19

References:

- "Flutter for Beginners" by Alessandro Biessek
- "Beginning Flutter: A Hands-On Guide to App Development" by Marco L. Napoli

Outcome: After completion of course, student will be able to:

CO1: Understand various mobile application architectures, design principles, and development patterns specific to Flutter.
CO2: Describe the components, structure, and functionalities of the Flutter framework and the Dart programming language.
CO3: Develop cross-platform mobile applications using Flutter and its core features.
CO4: Deploy, test, and debug Flutter applications on both Android and iOS platforms

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO5/PSO1
CO2	PO1,PO3/PSO1, PSO2
CO3	PO1,PO3/PSO2, PSO4
CO4	PO1/PSO1

BCAE0472: MOBILE APPLICATION DEVELOPMENT LAB II

Introduction:This lab deals with the development of mobile application.

Objective:The objective of this lab to provide the knowledge to develop the trendy mobile applications in various technologies (android and ios).

Credits: 02 **L–T–P: 0–0–4**

List Of Practical

Exp. No.	Content	Teaching Hours
1	Set up the Flutter environment, install the SDK, and configure the IDE.	48
2	Learn Dart basics (syntax, variables, functions).	
3	Create a "Hello World" app using Flutter's core widgets and display simple text.	
4	Build a user interface with ListView to display a list of items dynamically.	
5	Implement navigation between screens with arguments passed using Navigator.push().	
6	Design a responsive layout using Flex, Expanded, and Wrap widgets.	
7	Use TextField to accept user input and display it dynamically on the screen.	
8	Integrate Firebase Authentication for user sign-up and login functionality.	
9	Implement an interactive form with validation and submit functionality.	
10	Fetch and display data from a public API using http package in Flutter.	
11	Implement location services to fetch and display user location using the geolocator package.	

***Intended Outcomes*:** After completion of this course students will be able to:

- CO1-Students will be able to design and implement graphical user interfaces (GUIs) using various GUI components, fonts, and colours effectively to enhance the visual appeal of the application.
- CO2-Students will gain proficiency in using layout managers to organize GUI components effectively and understand the concept of event-driven programming. They will be able to implement event listeners to handle user interactions.
- CO3-Students will learn to develop a fully functional calculator application with basic arithmetic operations. Also, they will understand the implementation of GUI components, event listeners, and basic mathematical operations.
- CO4-Students will be able to implement multi-threading to perform tasks concurrently and improve application performance.
- CO5-Students will learn how to access GPS location information in a native application. They will be able to use location services and implement location-based features effectively.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO3/PSO2
CO2	PO3/PSO2
CO3	PO5,PO8/PSO1,PSO2,PSO4

CO4	P02, PO4, PO9/PSO1, PSO3
CO5	PO3, PO5/ PSO4