

<b>COURSE TITLE</b>	<b>MOBILE APPLICATION DEVELOPMENT WITH FLUTTER</b>
<b>COURSE CODE</b>	<b>01CC0602</b>
<b>COURSE CREDITS</b>	<b>4</b>

**Objective:**

- 1 This course focuses on learning and applying the fundamentals of Flutter Development to build mobile applications by learning the concepts of Dart, Widgets and Gestures, UI and List View, and Data Storage which will be compatible with cross platforms students , will be capable of designing, testing, and deploying production-ready applications to the Google Play Store

**Course Outcomes:** After completion of this course, student will be able to:

- 1 Apply dart OOP concepts, including inheritance, polymorphism, and asynchronous programming, to develop efficient and null-safe application logic
- 2 Apply layout and styling techniques to design responsive and visually appealing user interfaces using core Flutter widgets and MediaQuery
- 3 Design and develop different widget to design cross platform mobile application.
- 4 Evaluate different approaches for handling state management, user input, form validation to create interactive mobile apps.
- 5 Design and develop a complete Flutter application integrating UI components, state management, API services, local storage, and deployment to production.

**Pre-requisite of course:NA**

**Teaching and Examination Scheme**

<b>Theory Hours</b>	<b>Tutorial Hours</b>	<b>Practical Hours</b>	<b>ESE</b>	<b>IA</b>	<b>CSE</b>	<b>Viva</b>	<b>Term Work</b>
3	0	2	50	30	20	25	25

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
1	<b>Introduction to Flutter &amp; Dart</b> Dart Essentials: Variables, null safety, and Arrow functions., Object-Oriented Programming: Classes, Constructors, Inheritance, Polymorphism, Encapsulations,, Collections: List, Map, Set. Environment Setup: Flutter SDK, Android Studio configuration, and the flutter doctor utility, The "Everything is a Widget" Concept: Understanding the Widget tree and the main.dart entry point.	7

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
2	<b>Building Static UIs with Layouts</b> Types of Flutter Widgets, Stateless and Stateful Widget lifecycle, Working of Widget tree and Element tree, Core Layout Widgets: Container, Column, Row, Stack, and Center, Styling & Assets: Integrating Google Fonts, custom hex colors, and asset management in pubspec.yaml, Text, Image, Icon, Padding, Align, Center, SizedBox, Expanded, hot reload / hot restart workflow , MaterialApp, Scaffold, AppBar structure, Project: Building a Digital Business Card or Profile UI.	8
3	<b>Widgets fundamentals &amp; Responsive Layouts</b> Button widgets(Text button , Elevated Button , Outlined button , Icon button, Floating Action button, Toggle button , Segment button),badge, Snack bar, bottom navigation bar, Navigation bar, Checkbox, Radio button, Navigation rail, Date picker, Time picker, switch, Textfield, using Decorators, Using the Form Widget to Validate Text Fields, Gestures, Responsiveness: Using MediaQuery, Expanded, Flexible, and Aspect ratio for cross-device support.	9
4	<b>State Management &amp; Working with Lists</b> State Management: Understanding Stateful vs Stateless widgets and the State lifecycle, User Input: Handling TextField data, Form validation, and Button interactions, Data Flow: Concepts of "Lifting State Up" to manage parent-child communication, Navigation: Screen transitions using the Navigator (Push/Pop), Scrollable Widgets: ListView.builder, GridView.builder, and SingleChildScrollView, Refactoring: Extracting widgets into reusable custom components, Advanced Components: Implementing ListTile, Card, SliverAppBar, and Navigation Drawers.	9
5	<b>Working with Networking , API integration , Data Storage &amp; Deployment</b> Asynchronous Programming: Working with Future, async, and await, HTTP Integration: Fetching and parsing JSON data from REST APIs. , SQLite database: Introduction to SQLite, CRUD operations. Persistence: Saving user preferences using shared preferences, Animations: Implementing Hero transitions and implicit animations, Deployment: Building the App Bundle/APK and a walkthrough of the Play Store submission process	9
<b>Total Hours</b>		<b>42</b>

#### Suggested List of Experiments:

<b>Contents : Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
1	<b>Practical 1</b> Install Flutter SDK and Android Studio, create a new Flutter project named recipe_book_app, and design a splash screen with text and app logo	2

### Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
2	<b>Practical 2</b> Create a Home Screen containing three buttons: All Recipes, Add Recipe, and Categories.	2
3	<b>Practical 3</b> Display recipe categories such as Indian, Italian, Chinese, Mexican, and Desserts using icons in a grid layout.	2
4	<b>Practical 4</b> Build a form with fields for Recipe Name, Ingredients, Steps, Category, Time Required, and optional image upload, including proper form validation.	2
5	<b>Practical 5</b> Implement named routes to navigate between Home, Add Recipe, and Recipe List screens.	2
6	<b>Practical 6</b> Display a dummy recipe list showing title, category, and cooking time.	2
7	<b>Practical 7</b> Build a Recipe Detail Page showing ingredients, preparation steps, and include Edit and Delete buttons.	2
8	<b>Practical 8</b> Create a recipe card widget displaying image, recipe title, and category, and reuse it across the app.	2
9	<b>Practical 9</b> Create a Recipe model and write code to insert recipe form data into SQLite.	2
10	<b>Practical 10</b> Display saved recipes from SQLite, edit existing recipes, delete recipes, and update the ListView accordingly.	2
11	<b>Practical 11</b> Use TheMealDB API to fetch trending dessert recipes, parse the JSON response, and display data in ListView or GridView.	2
12	<b>Practical 12</b> Add a search bar on the All Recipes page and dynamically filter recipes based on user input.	2
13	<b>Practical 13</b> Add favorite icon toggle for recipes, update UI dynamically, and display all favorites on a separate page.	2
<b>Total Hours</b>		<b>26</b>

### Textbook :

- 1 Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, Rap Payne, Springer, 2019

### References:

- 1 Flutter for Beginners, Flutter for Beginners, Flutter for Beginners, Flutter for Beginners, Thomas Bailey and Alessandro Biessek, Packt Publishing, 2021
- 2 Google Flutter Mobile Development Quick Start Guide, Google Flutter Mobile Development Quick Start Guide, Google Flutter Mobile Development Quick Start Guide, Google Flutter Mobile Development Quick Start Guide, Prajyot Mainkar and Salvatore Girodano, Packt Publishing, 2019

### Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
0.00	10.00	40.00	30.00	20.00	0.00

### Instructional Method:

- 1 The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc
- 2 The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
- 3 Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.
- 4 Students will use supplementary resources such as online videos, NPTEL videos, ecourses, Virtual Laboratory.

### Supplementary Resources:

- 1 <https://dart.dev/guides>
- 2 <https://docs.flutter.dev/>
- 3 <https://www.youtube.com/@flutterdev>
- 4 <https://medium.com/flutter>