

## SEMESTER S6

### MOBILE APPLICATION DEVELOPMENT

(Common to CS/CA/CB/CN)

<b>Course Code</b>	<b>PECST695</b>	<b>CIE Marks</b>	40
<b>Teaching Hours/Week (L: T:P: R)</b>	3:0:0:0	<b>ESE Marks</b>	60
<b>Credits</b>	5/3	<b>Exam Hours</b>	2 Hrs. 30 Min.
<b>Prerequisites (if any)</b>	None	<b>Course Type</b>	Theory

#### Course Objectives:

1. To equip students with a thorough understanding of mobile application development fundamentals, including platforms (iOS and Android) and architectures (MVC, MVVM, BLoC).
2. To instill proficiency in Flutter and Dart: Enable students to use Flutter effectively for cross-platform development and the Dart programming language to create responsive, user-friendly mobile applications.
3. To prepare students for real-world scenarios by teaching app security, testing, CI/CD, and deployment processes, culminating in the development and deployment of a complete mobile application project.

### SYLLABUS

<b>Module No.</b>	<b>Syllabus Description</b>	<b>Contact Hours</b>
<b>1</b>	<b>Fundamentals of Mobile Application Development:</b> Introduction to Mobile Application Development, Overview of Mobile Platforms: iOS and Android, Introduction to Flutter: History, Features, and Benefits, Setting Up the Flutter Development Environment, Mobile App Architectures (MVC, MVVM, and BLoC), Basics of Dart Programming Language, Introduction to Git and Version Control <b>Assignments/Projects:</b> Set up the Flutter environment and create a simple "Hello World"	<b>9</b>

	<p>application. (<i>Use Git: cloning, committing, pushing, and pulling</i>)</p> <p><i>Milestone 1:</i> Develop a basic app with a simple UI and basic functionality.</p>	
2	<p><b>User Interface Design and User Experience:</b></p> <p>Principles of Mobile UI/UX Design, Designing Responsive UIs with Flutter, Using Flutter Widgets: StatelessWidget and StatefulWidget, Layouts in Flutter: Container, Column, Row, Stack, Navigation and Routing in Flutter, Customizing UI with Themes and Styles, Introduction to Material Design and Cupertino Widgets</p> <p><b>Assignments/Projects:</b></p> <p>Design and implement a user interface using Flutter widgets.</p> <p><i>Milestone 2:</i> Enhance the project from Module 1 with a multi-screen UI, navigation, and customized themes.</p>	9
3	<p><b>Advanced Flutter Development:</b></p> <p>State Management in Flutter: Provider, Riverpod, and BLoC</p> <p>Networking in Flutter: HTTP Requests, JSON Parsing, RESTful APIs</p> <p>Data Persistence: SQLite, SharedPreferences, Hive</p> <p>Asynchronous Programming with Dart: Futures, async/await, and Streams</p> <p>Integrating Device Features: Camera, GPS, Sensors</p> <p>Working with Firebase: Authentication, Firestore, Cloud Functions</p> <p><b>Assignments/Projects:</b></p> <p>Develop an app with state management and data persistence.</p> <p><i>Milestone 3:</i> Enhance the project with state management, data persistence, and integration with a RESTful API or Firebase.</p>	9
4	<p><b>Industry Practices and App Deployment:</b></p> <p>Advanced UI Components and Animations, App Security Best Practices, Testing and Debugging Flutter Applications, Continuous Integration/Continuous Deployment (CI/CD) with Flutter, Publishing Apps to Google Play Store and Apple App Store, Industry Trends and Future of Mobile Development with Flutter</p> <p><b>Assignments/Projects:</b></p> <p>Add advanced UI components and animations to the project, Implement</p>	9

	<p>security measures in the Flutter application, Conduct thorough testing and debugging of the developed app.</p> <p><i>Milestone 4:</i> Complete the project, integrating all features and preparing it for deployment.</p>	
--	--	--

**Course Assessment Method**  
**(CIE: 40 marks, ESE: 60 marks)**

**Continuous Internal Evaluation Marks (CIE):**

<i>Attendance</i>	<i>Internal Ex</i>	<i>Evaluate</i>	<i>Analyse</i>	<i>Total</i>
<b>5</b>	<b>15</b>	<b>10</b>	<b>10</b>	<b>40</b>

**Criteria for Evaluation(Evaluate and Analyse): 20 marks**

**Analyze**

Key Actions: Differentiate, Organize, Attribute

Metrics and Examples:

1. Code Review and Refactoring:
  - a. Task: Students are given a piece of code to analyze and refactor for better performance or readability.
  - b. Metric: Ability to identify inefficient or redundant code and provide optimized solutions.

Example: Analyzing a complex UI widget tree and reorganizing it for better performance and maintainability.

2. Design Pattern Identification:
  - a. Task: Students are asked to identify and apply appropriate design patterns for given scenarios.
  - b. Metric: Correct identification and application of design patterns like Singleton, Factory, or BLoC in their projects.

Example: Analyzing an app's state management needs and choosing between Provider and BLoC patterns.

3. Bug Diagnosis:

- a. Task: Students are given a buggy piece of code to analyze and debug.
- b. Metric: Ability to use debugging tools and techniques to locate and fix bugs.

Example: Analyzing asynchronous code to identify and resolve race conditions or memory leaks.

## **Evaluate**

Key Actions: Check, Critique, Judge

Metrics and Examples:

1. Code Quality Assessment:

- a. Task: Students review each other's code and provide constructive feedback.
- b. Metric: Ability to critically evaluate code quality based on readability, efficiency, and adherence to best practices.

Example: Peer review sessions where students critique the structure and efficiency of each other's Flutter code.

2. UI/UX Design Evaluation:

- a. Task: Students evaluate the user interface and user experience of their peers' applications.
- b. Metric: Ability to judge UI/UX designs based on usability, accessibility, and aesthetics.

Example: Conducting usability testing sessions and providing feedback on navigation flow, design consistency, and user engagement.

3. Project Presentation and Defense:

- a. Task: Students present their projects and justify their design and implementation choices
- b. Metric: Ability to articulate design decisions, defend architectural choices, and respond to critical questions.

Example: End-of-module presentations where students explain their choice of state management, navigation strategy, and performance optimizations.

## **Integration into the Syllabus - Example Use Cases**

Basic Mobile Application Development

- Analyze: Evaluate different mobile app architectures (MVC, MVVM, BLoC) and choose the best fit for a given project scenario.
- Evaluate: Critically assess the setup and configuration of the Flutter development environment for potential improvements.

#### User Interface Design and User Experience

- Analyze: Analyze the responsiveness and usability of designed UIs, identifying potential bottlenecks.
- Evaluate: Critique the effectiveness of navigation and routing within the app.

#### Advanced Flutter Development

- Analyze: Break down the integration process of advanced features (state management, networking) and evaluate their impact on app performance.
- Evaluate: Judge the robustness of data persistence solutions and asynchronous programming implementations.

#### Industry Practices and App Deployment

- Analyze: Analyze the app's security measures and their effectiveness in protecting user data.
- Evaluate: Evaluate the completeness and readiness of the app for deployment based on industry standards and best practices.

### Example Evaluation Rubrics

#### Analyze:

Criterion	Excellent (4)	Good (3)	Satisfactory (2)	Needs Improvement (1)
Identification of Code Inefficiencies	Identifies all inefficiencies and provides optimal solutions	Identifies most inefficiencies and provides good solutions	Identifies some inefficiencies with basic solutions	Struggles to identify inefficiencies or provide solutions
Application of Design Patterns	Correctly applies design patterns with a clear rationale	Applies design patterns with minor issues	Applies design patterns with significant issues	Incorrectly applies or fails to apply design patterns

**Evaluate:**

<b>Criterion</b>	<b>Excellent (4)</b>	<b>Good (3)</b>	<b>Satisfactory (2)</b>	<b>Needs Improvement (1)</b>
Code Quality Assessment	Provides thorough, insightful feedback with constructive suggestions	Provides good feedback with some constructive suggestions	Provides basic feedback with limited constructive suggestions	Provides minimal or unhelpful feedback
UI/UX Design Evaluation	Provides detailed critique with actionable insights	Provides good critique with some actionable insights	Provides basic critique with limited actionable insights	Provides minimal or no critique

**End Semester Examination Marks (ESE):**

*In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions*

<b>Part A</b>	<b>Part B</b>	<b>Total</b>
<ul style="list-style-type: none"> <li>• 2 Questions from each module.</li> <li>• Total of 8 Questions, each carrying 3 marks <b>(8x3 =24marks)</b></li> </ul>	2 questions will be given from each module, out of which 1 question should be answered. Each question can have a maximum of 3 sub divisions. Each question carries 9 marks. <b>(4x9 = 36 marks)</b>	<b>60</b>

### Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
<b>CO1</b>	Explain mobile application development using Flutter and different mobile platforms.	<b>K2</b>
<b>CO2</b>	Apply principles of effective mobile UI/UX design, Create responsive user interfaces using Flutter features.	<b>K3</b>
<b>CO3</b>	Experiment effectively with state in Flutter application, networking and data persistence.	<b>K4</b>
<b>CO4</b>	Apply security best practices in mobile app development, test, and debug Flutter applications effectively.	<b>K5</b>
<b>CO5</b>	Set up CI/CD pipelines for Flutter projects and deploy mobile apps to Google Play Store and Apple App Store.	<b>K5</b>

Note: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create

### CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	3	3	3	3								3
<b>CO2</b>	3	3	3	3	3							3
<b>CO3</b>	3	3	3	3	3							3
<b>CO4</b>	3	3	3	3	3							3
<b>CO5</b>	3	3	3	3	3							3

Note: 1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation

Text Books				
Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
1	Flutter Cookbook	Simone Alessandria	Packt	2/e, 2023
2	Flutter for Beginners	Alessandro Biessek	Packt	1/e, 2019

<b>Reference Books</b>				
<b>Sl. No</b>	<b>Title of the Book</b>	<b>Name of the Author/s</b>	<b>Name of the Publisher</b>	<b>Edition and Year</b>
1	Flutter in Action	Eric Windmill	Manning	1/e, 2019
2	Flutter and Dart: Up and Running	Deepti Chopra, Roopal Khurana	BPB	1/e, 2023
3	Managing State in Flutter Pragmatically	Waleed Arshad	Packt	1/e, 2021
4	Ultimate Flutter Handbook	Lahiru Rajeendra Mahagamage	Orange House	1/e, 2023

<b>Video Links (NPTEL, SWAYAM...)</b>	
<b>No.</b>	<b>Link ID</b>
<b>1</b>	<a href="https://www.youtube.com/watch?v=VPvVD8t02U8">https://www.youtube.com/watch?v=VPvVD8t02U8</a>