303105380 - Mobile App Development Laboratory

Course: Btech **Semester:** 6

Prerequisite: Basic knowledge of java language

Rationale : The mobile application development syllabus covers the essential concepts and tools for building apps across platforms, including UI/UX design, app architecture, networking, databases, and deployment. It explores both native development (Android) and cross-platform frameworks, emphasizing practical skills for creating functional, user-friendly mobile applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Acquire an insight into concepts of mobile application development terminologies, environment and architecture 2. Design mobile application using various UI components and layouts. 3. Develop robust mobile applications with database interaction and webservice integration 4. Deploy application on mobile device.

303105380 - Mobile App Development Laboratory

List of Practical

1.	**Create a "Hello World" application: Display "Hello World" at the center of the screen, both on the Android emulator and an actual Android device.**
2.	**Build an app to showcase Android lifecycle phases: Develop an app that demonstrates various Android lifecycle stages (onCreate, onStart, onResume, etc.).**
3.	**Create an app with two activities: The first activity should contain an EditText and a "Send" button. When the button is clicked, use an explicit intent to send the text from EditText to a second activity and display it in a TextView.**
4.	**Create an app with explicit intent: The first activity should have an EditText and a "Send" button. On button click, use an implicit intent with the "SEND" action, allowing the user to select an app from an app chooser to handle the intent and display the text.**

5.	Build a basic calculator app: Create an app that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on numbers.
6.	Create a Spinner-based app: Develop an app with a spinner populated from the res/values/strings.xml resource. When the spinner value changes, the corresponding image from the res/drawable directory should be displayed.
7.	Create a discount calculator app: Use a RadioGroup with three radio buttons for 10%, 15%, and 20% discounts on a shopping bill. The user can enter the bill amount in an EditText, and the selected discount will be calculated and displayed in a TextView.
8.	Create an app with a course selection RadioButton group: Display a list of college courses with a RadioButtongroup. When a course is selected, the corresponding TIC (Total Instructional Credit) should be shown in a TextView.
9.	Create a shopping list app using checkboxes: Build an app with checkboxes for shopping list items. As items are checked off, the selected items should be displayed in a TextView.
10.	Create a login and registration app: Develop a login application that verifies the username and password. Include a registration page for new users. Upon successful login, show a "Welcome User" pop-up message
11.	Create a login app with navigation to another activity: The login screen should verify the username and password. After successful login, navigate to a new activity that displays a "Welcome User" message in a TextView and a "Logout" button. On clicking "Logout," show a confirmation dialog with "OK" and "Cancel" buttons. "OK" should return to the login screen, while "Cancel" should keep the user on the current activity.
12.	Create an app with a menu: Implement a menu with five options. The selected option should be displayed in a TextView.
13.	Build an app using LinearLayout: Create a simple app that uses LinearLayout. It should take the contents of a predefined TextView, convert it to uppercase on button click, and display it in an EditText. Additionally, create an app that responds to key events in the EditText without needing a button press.
14.	Create an app with TableLayout and custom styles: Use a TableLayout with a TextView, EditText, and buttons. Also, create a custom styles.xml in the res/values directory to style the TextView.
15.	Create an app with SQLite database operations: Build an app that allows the user to perform CRUD operations (Create, Read, Update, Delete) with an SQLite database.
16.	Create an app with three vertically aligned buttons: Develop an app with three buttons arranged vertically. When any button is selected, the screen color should change accordingly.