

COMP 019 - Applications Development and Emerging Technologies

Full-Stack Python Development

SESSION 4: MOBILE APP DEVELOPMENT WITH KIVY

ACTIVITY 4.1: KIVY SETUP AND BASICS

Topic: Introduction to Kivy framework for mobile apps

Description: Set up Kivy and create your first mobile application

INSTRUCTIONS:

Install and verify Kivy:

- pip install kivy
- Create test file and run

Understand Kivy basics:

- App class - main application
- Widget - UI components
- Layouts - organize widgets
- Properties - reactive data binding

Create first Kivy app:

- Create class extending App
- Override build() method
- Return root widget
- Run with python main.py

Create simple UI:

- Label widget with text
- Button widget with on_press callback
- Display message when button clicked

Document Kivy vs web development differences

Deliverables: Submit Kivy app code and running app screenshots

25 Points

ACTIVITY 4.2: KIVY LAYOUTS AND WIDGETS

Topic: Building user interfaces with Kivy layouts

Description: Master Kivy layouts and common widgets for mobile UI

INSTRUCTIONS:

Learn Kivy layouts:

- BoxLayout - horizontal or vertical stacking
- GridLayout - rows and columns
- FloatLayout - absolute positioning
- StackLayout - flow layout

Common widgets:

- Label - display text
- Button - clickable button
- TextInput - text entry field
- Image - display images
- Slider, CheckBox, Switch

Create a login screen:

- Logo/title at top
- Username TextInput
- Password TextInput (password=True)
- Login Button
- 'Register' link/button

Style with Kivy properties:

- size_hint, pos_hint
- Colors, fonts, padding

Deliverables: Submit login screen code and UI screenshot

30 Points

ACTIVITY 4.3: KIVY LANGUAGE (KV FILES)

Topic: Separating UI design with KV language

Description: Use KV language for cleaner, maintainable UI code

INSTRUCTIONS:

Understand KV language:

- Separate UI design from Python logic
- Automatic binding with ids
- Cleaner, more readable code

Create .kv file:

- Name must match App class (e.g., MyApp → my.kv)
- Define widget tree
- Use indentation for hierarchy

KV language syntax:

- Widget properties: text: 'Hello'
- Event binding: on_press: root.my_function()
- Dynamic properties: text: self.parent.title
- Referencing widgets: id: my_button

Build complete app with KV:

- Main screen with navigation
- List screen showing items
- Detail screen for single item
- Use ScreenManager for navigation

Deliverables: Submit Python and KV files with multi-screen app demo

35 Points

TOTAL POINTS: 90