Opens: Friday, November 28, 2025, 10:00 AM
Due: Friday, November 28, 2025, 12:00 PM

---

1. Objective

Create a single-screen Jetpack Compose application that demonstrates your understanding of:

- **State Management:** Using `remember` and `mutableStateOf` to handle data changes.
- **TextField & Text:** capturing user input and displaying output.
- **Kotlin String Manipulation:** Processing text logic in real-time.

2. UI Requirements

Please build a screen that includes the following elements, arranged vertically (using a `Column`):

1. **Input Area (Top):**
    - A `TextField` (or `OutlinedTextField`) where the user can type text.
    - It should have a placeholder or label saying "Enter text here".
    - It should fill the width of the screen (`fillMaxWidth`).
2. **Display Area (Bottom):**
    - A `Text` component displaying the processed result.
    - The text size should be larger (e.g., `30.sp` or `MaterialTheme.typography.h4`) to be clearly visible.
    - The text color should be different from the input (e.g., Blue or Purple).

3. Logic Requirements

As the user types in the input area, the Display Area must update **in real-time** following these rules:

- **Title Case Transformation:** For every word, the **first letter must be Capitalized**, and all subsequent letters in that word must be **lowercased**.
    - *Example:* Input `aNdRoId` -> Output `Android`
- **Mixed Input:**
    - *Example:* Input `jEtPaCk cOmPoSe` -> Output `Jetpack Compose`
- **Symbols & Spaces:** Do not modify spaces, punctuation, or numbers. They should appear exactly as typed.

- *Example:* Input `hello, world!` -> Output `Hello, World!`

4. Hint

- You will need to hoist the state. The `value` variable should be stored outside the TextField so it can be passed to the transformation logic for the Text view.
- You might find Kotlin's standard library functions like `.split()`, `.lowercase()`, and `.replaceFirstChar { ... }` useful.

5. Visual Reference

See the mockup below for the expected layout.

In-Class: The Real-time Title Case Converter App

1. Objective Create a single-screen Jetpack Compose application that demonstrates your understanding of: State Management: Using remember and mutableStateOf to handle data changes. TextField & Text: capturing user input and displaying output. Kotlin String Manipulation: Processing text logic in real-time. 2. UI Requirements Please build a screen that includes the following elements, arranged vertically (using a Column): Input Area (Top): A TextField (or OutlinedTextField) where the user can type text. It should have a placeholder or label saying "Enter text here". It should fill the width of the screen (fillMaxWidth). Display Area (Bottom): A Text component displaying the processed result. The text size should be larger (e.g., 30.sp or MaterialTheme.typography.h4) to be clearly visible. The text color should be different from the input (e.g., Blue or Purple). 3. Logic Requirements As the user types in the input area, the Display Area must update in real-time following these rules: Title Case Transformation: For every word, the first letter must be Capitalized, and all subsequent letters in that word must be lowercased. Example: Input aNdRoId -> Output Android Mixed Input: Example: Input jEtPaCk cOmPoSe -> Output Jetpack Compose Symbols & Spaces: Do not modify spaces, punctuation, or numbers. They should appear exactly as typed. Example: Input hello, world! -> Output Hello, World! 4. Hint You will need to hoist the state. The value variable should be stored outside the TextField so it can be passed to the transformation logic for the Text view. You might find Kotlin's standard library functions like .split(), .lowercase(), and .replaceFirstChar { ... } useful. 5. Visual Reference See the mockup below for the expected layout.

Group	ASEP STROBERI V2
Submission status	Nothing submitted yet. Not editable because: Assessment not open yet. Users who did not submit: 412855503 曾志偉, 412416074 李富豪, 412855149 林樂安, 412855214 薛志強, 412855024 黃庭豪, 412856139 黃輝宏

Due date	Friday, November 28, 2025, 12:00 PM
Time remaining	2 days 19 hours