

## Interactive On-Screen Keyboard and GUI Development

In this project, students will design and implement a functional on-screen keyboard interface. This project serves as a foundational step in developing assistive technologies, focusing on graphical user interface (GUI) design, event-driven programming, and real-time visual feedback mechanisms.

### Key Objectives

- **GUI Development:** Build a simple, organized layout for an on-screen keyboard using Python libraries such as Tkinter or PyQt.
- **Input Interaction:** Enable the user to interact with the keyboard by using a mouse to press buttons and input characters.
- **Visual State Feedback:** Implement dynamic color changes to keys based on user interaction to provide clear visual confirmation.

### Technical Tasks

#### 1. Keyboard Layout Design:

- Construct a grid or flexible layout to represent a standard keyboard.
- Ensure each key is an interactive button object that can be addressed individually.

#### 2. Mouse Interaction Logic:

- Map mouse click events to the corresponding character input for each button.

#### 3. Dynamic Visual Effects:

- **Hover State:** Implement a "mouseover" event that changes the background color of a key when the cursor moves over it to highlight the current target.
- **Active State:** Implement a "pressed" event that changes the key's color temporarily when clicked to simulate a physical button press.

### Deliverables

- **Software Application:** A working GUI keyboard where users can "type" using a mouse.
- **Source Code:** Documented Python script demonstrating the implementation of event listeners and GUI styling.

## **Learning Outcomes**

**By completing this project, students will:**

- Gain practical experience in GUI development and event-driven programming.
- Understand how to provide immediate visual feedback to improve the user experience in assistive software.