**Data Flow Diagram (DFD) for Theme Context App**

**Theme Context App** allows users to toggle between **light mode** and **dark mode** using React's **Context API**. Below is the **DFD representation**:

**Level 0 (Context Diagram)**

At the highest level, the system consists of:

1. **External Entity: User** → Clicks the button to toggle the theme.
2. **Process: Theme Context System** → Manages and updates the theme across the app.
3. **Data Store: React Context (ThemeContext)** → Stores the current theme and updates state.

+--------------------+

| External Entity: |

| User |

+--------------------+

|

v

+--------------------+

| Theme Context | (Process 1.0)

| System |

+--------------------+

|

v

+--------------------+

| ThemeContext | (Stores theme state)

| useState Hook |

+--------------------+

**Level 1 DFD (Decomposed Processes)**

Breaking down **Process 1.0 (Theme Context System)** into detailed steps:

+--------------------+ +------------------------+

| External Entity: | | Process 1.1 - User |

| User | -----> | Clicks Toggle Button |

+--------------------+ +------------------------+

| |

v v

+--------------------+ +------------------------+

| Process 1.2 - | | Process 1.3 - Update |

| Capture Event | -----> | (Change Theme State) |

+--------------------+ +------------------------+

| |

v v

+--------------------+ +------------------------+

| Process 1.4 - | | Process 1.5 - Apply |

| Store Theme | -----> | (Change UI Theme) |

| (useState Hook) | | (Light/Dark Mode) |

+--------------------+ +------------------------+

**Explanation of Level 1 DFD Processes**

1. **Process 1.1 - User Clicks Toggle Button**
   * The user clicks the **"Toggle Theme"** button.
2. **Process 1.2 - Capture Event**
   * The button click triggers an event handled by the ThemeSwitcher component.
3. **Process 1.3 - Update Theme State (useState)**
   * The setTheme function updates the theme between "light" and "dark".
4. **Process 1.4 - Store Theme in Context (ThemeContext)**
   * The updated theme is stored in ThemeContext so that it can be used across the app.
5. **Process 1.5 - Apply Theme to UI**
   * The theme state is used to apply **light mode** or **dark mode** styles dynamically.

**Data Flow Summary**

✔ **User clicks a button → System captures event → Updates theme state → Applies new theme**  
✔ Uses **React Context (ThemeContext)** for global state management.  
✔ **No backend or database**—only front-end state management with Context API.