**Data Flow Diagram (DFD) for Simple Stopwatch App**

**Stopwatch App** allows users to start, stop, and reset a timer while also switching between **light mode** and **dark mode**. Below is the **DFD representation**:

**Level 0 (Context Diagram)**

At the highest level, the system consists of:

1. **External Entity: User** → Clicks buttons to start, stop, or reset the stopwatch.
2. **Process: Stopwatch System** → Controls timer functionality and updates the display.
3. **Data Store: React State (useState Hook)** → Stores elapsed time, running state, and theme mode.

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

| External Entity: |

| User |

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

|

v

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

| Stopwatch System | (Process 1.0)

| (React & useState) |

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

|

v

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

| useState Hook | (Stores time, running status, dark mode)

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

**Level 1 DFD (Decomposed Processes)**

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

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

| External Entity: | | Process 1.1 - Start |

| User | -----> | (Begin Timer) |

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

| |

v v

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

| Process 1.2 - | | Process 1.3 - Stop |

| Update Timer | -----> | (Pause Timer) |

| (useEffect Hook) | | (Clear Interval) |

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

| |

v v

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

| Process 1.4 - | | Process 1.5 - Reset |

| Display Timer | -----> | (Reset Time) |

| (Show Time) | | (Set time = 0) |

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

|

v

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

| Process 1.6 - |

| Toggle Theme |

| (Light/Dark Mode)|

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

**Explanation of Level 1 DFD Processes**

1. **Process 1.1 - Start Timer**
   * User clicks "Start", setting running state to true.
   * useEffect starts an interval that updates the time every second.
2. **Process 1.2 - Update Timer (useEffect)**
   * Every second, setTime updates the stored time in useState.
3. **Process 1.3 - Stop Timer**
   * User clicks "Stop", setting running state to false.
   * useEffect clears the interval to pause the timer.
4. **Process 1.4 - Display Timer**
   * The updated time is displayed dynamically in the UI.
5. **Process 1.5 - Reset Timer**
   * User clicks "Reset", setting time to 0 and running to false.
6. **Process 1.6 - Toggle Theme**
   * User clicks the "Theme Toggle" button.
   * The darkMode state is toggled, updating the background color and styles.

**Data Flow Summary**

✔ **User interacts → Timer updates state → System displays time and updates UI**  
✔ Uses **React state (useState) and effects (useEffect)** for real-time updates.  
✔ **No backend or database**—only front-end state management.