**Level 0 (Context Diagram)**

At the highest level, the **Theme Switcher System** allows the user to toggle between light and dark themes.

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

| External Entity |

| (User) |

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

|

v

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

| Theme Switcher App |

| (Process: 1.0) |

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

|

v

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

| Data Store: Theme |

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

**Explanation:**

* **External Entity (User):** The user interacts with the app by toggling the theme.
* **Process (Theme Switcher App):** The system processes the user’s request and updates the theme.
* **Data Store (Theme):** Stores the current theme state (light or dark).

**Level 1 DFD (Decomposition of Process 1.0 - Theme Switcher App)**

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

| External Entity |

| (User) |

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

|

v

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

| Process: 1.1 - Click Button |

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

|

v

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

| Process: 1.2 - Toggle Theme |

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

|

v

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

| Data Store: Theme |

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

**Explanation:**

1. **Process 1.1 (Click Button):**
   * The user clicks the "Toggle Theme" button.
2. **Process 1.2 (Toggle Theme):**
   * The system checks the current theme.
   * If **light**, it switches to **dark**.
   * If **dark**, it switches to **light**.
3. **Data Store (Theme):**
   * The app stores the updated theme.

**Data Flow**

* **User** interacts with the **button**.
* The **ThemeContext** processes the request.
* The system **updates** and **stores** the selected theme.
* The **UI** updates based on the **new theme**.

**Additional Notes:**

* Uses **React Context API** for state management.
* The ThemeProvider wraps the app and manages the theme.
* Future Enhancements:
  + **Persistent Theme Storage** (e.g., Local Storage).
  + **More Themes** (e.g., Blue, Green, Custom).
  + **Animations** for smooth transitions.