**Data Flow Diagram (DFD) for Themed Lazy Loading App**

**Project Description**

The Themed Lazy Loading App is a React-based web application that supports theme switching between light and dark mode. It also uses lazy loading to dynamically load About and Contact sections only when needed, optimizing performance. The system processes user input, manages the application state dynamically, and updates the UI accordingly.

**Level 0 DFD (Context Diagram)**

The Level 0 DFD provides a high-level overview of the system, illustrating how users interact with the Themed Lazy Loading App and how data flows between various components.

+-----------------------+  
| External Entities |  
| |  
| - User |  
+-----------------------+  
 |  
 v  
+---------------------------+  
| Themed Lazy Loading App |  
| (Process: 1.0) |  
+---------------------------+  
 |  
 v  
+---------------------------+  
| Data Store: Application |  
| State (Theme & Components) |  
+---------------------------+

**Explanation:**

• **External Entity (User):** The user interacts with the application by toggling themes and navigating to different sections.

• **Process (Themed Lazy Loading App):** The system processes user input, manages theme changes, and dynamically loads components.

• **Data Store (Application State - Theme & Components):** Stores the current theme and tracks the loaded components.

**Level 1 DFD (Detailed Breakdown)**

The Level 1 DFD further decomposes the Themed Lazy Loading App process into sub-processes, showing how user interactions trigger different functionalities.

+-----------------------+  
| External Entity |  
| (User) |  
+-----------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.1 - Toggle Theme|  
| (Light/Dark Mode) |  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.2 - Lazy Load |  
| (Load About/Contact) |  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.3 - Update UI |  
| (Render Selected Theme & Components) |  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Data Store: Application |  
| State (Theme & Components) |  
+----------------------------+

**Explanation:**

1. **Process 1.1 (Toggle Theme - Light/Dark Mode):** The user clicks the toggle button, and the application switches between themes.

2. **Process 1.2 (Lazy Load - Load About/Contact):** When the user navigates, the application loads the required components dynamically.

3. **Process 1.3 (Update UI - Render Selected Theme & Components):** The UI updates based on the selected theme and dynamically loaded components.

**Data Flow**

• The User toggles the theme or navigates between sections.

• The System processes the input, updating the application state.

• The UI dynamically reflects the changes in theme and loaded components.

**Additional Notes**

• This system optimizes performance using React's lazy loading and Suspense.

• Future enhancements could include user preferences storage for theme persistence.