**DYNAMIC REACT PAGE**

This project is a **React web application** that implements **dynamic routing** using react-router-dom. It includes navigation between **Home, About, Contact, and Post pages**, and dynamically fetches post data from an external API (https://jsonplaceholder.typicode.com/posts/:id).

LEVEL 00: Context Diagram

* User : Navigates between the pages(Home,About,Contact,Post).
* System(React App) : Handles routing and fetches dynamic post data.

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

| User |

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

|

v

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

| React Routing System (0) |

| - Manages navigation |

| - Fetches dynamic post data |

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

|

v

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

| JSONPlaceholder API (External)|

| - Provides dynamic post data |

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

LEVEL 01 : Breaking down the system

* Navigation Handling(1.1) : Manages user navigation between pages.
* Dynamic Post Fetching(1.2) : Fetches post data dynamically from an API.
* Error Handling and Not Found Page(1.3) : Handles invalid routes.

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

| User |

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

|

v

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

| Process 1.1: Navigation |

| - Home, About, Contact |

| - Dynamic post routing |

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

|

v

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

| Process 1.2: Fetch Post |

| - Calls JSONPlaceholder API |

| - Displays post data |

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

|

v

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

| Process 1.3: Error Handling |

| - Invalid URLs → 404 Page |

| - Handles API errors |

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

**Process 1.1 : Navigation Handling**

* **(User Clicks Navigation Link)** – User clicks Home, About, or Contact.
* **(Route Matching)** – The system matches the URL and loads the respective component.

**Process 1.2 : Dynamic Post Fetching**

* **(User Visits /post/:id)** – The user navigates to a dynamic post page.
* **(Extract Post ID)** – The system extracts the post ID from the URL using useParams().
* **(Fetch Data from API)** – The app sends a request to <https://jsonplaceholder.typicode.com/posts/:id>.
* **(Display Post Details)** – The retrieved post is displayed with a title and body.

**Process 1.3 : Error Handling**

* **(Invalid Post ID)** – If the API returns an error, a message is displayed.
* **(Invalid URL Access)** – If a user enters an unknown URL, the system shows a **404 page**.

### **Data Flow**

1. The **user** requests a page.
2. If it's a **static page**, the system renders it.
3. If it's a **post page**, the system **fetches data** from an external API.
4. If the page **doesn’t exist**, the system returns the **404 page**.