Dynamic Page Application Data Flow

1. Introduction

This document provides an in-depth Data Flow Diagram (DFD) representation of the Dynamic Page React application.

It describes how navigation, dynamic routing, and data fetching are handled in the system.

2. System Overview

The React application consists of the following key components:

- App.js: The main file that sets up React Router for dynamic navigation.
- components/
  - Home.js: Displays the homepage.
  - About.js: Displays information about the application.
  - Contact.js: Displays contact details.
  - Post.js: Fetches and displays dynamic content based on route parameters.
  - NotFound.js: Displays a 404 error page for unknown routes.

3. Data Flow Diagram (DFD)

Level 0 (Context Diagram)

At a high level, the system consists of external users interacting with the application through a dynamic UI.

External Entities:

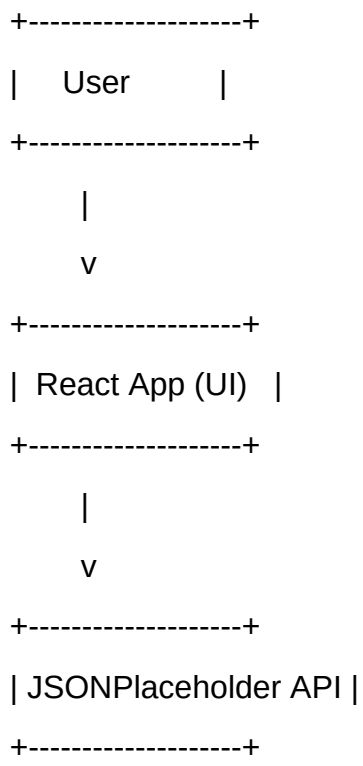
- User: Navigates between different pages using React Router.

Processes:

- React Application: Manages rendering, routing, and dynamic data fetching.

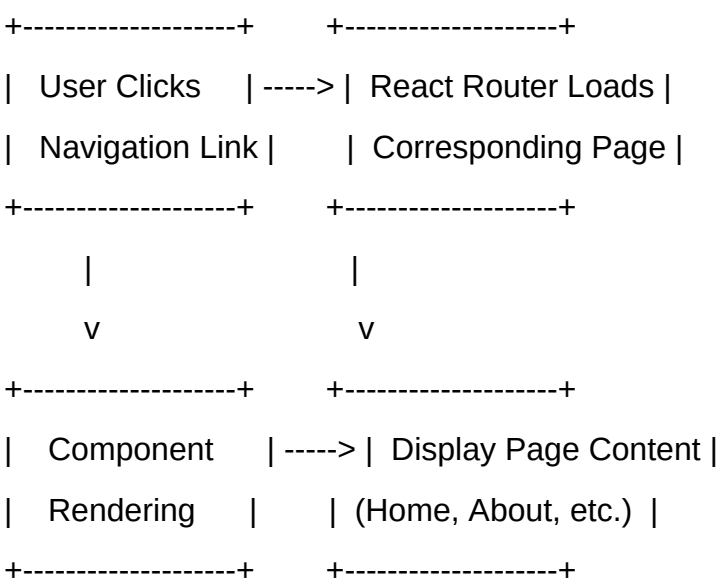
Data Stores:

- JSONPlaceholder API: Provides post data for the dynamic Post component.



Level 1 DFD (Navigation Process)

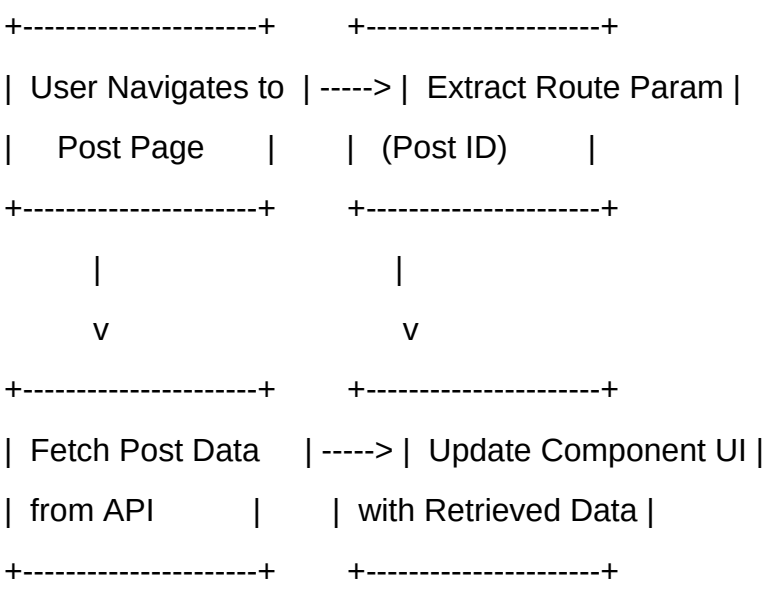
This level breaks down the navigation and routing mechanism.



Process Explanation:

1. The user interacts with the navigation bar and clicks a link.
2. React Router dynamically loads the corresponding component.
3. The application renders the component and displays its content.

Level 2 DFD (Dynamic Data Fetching in Post.js)



Process Explanation:

1. The user visits a post page (e.g., `/post/1`).
2. React Router extracts the `id` parameter from the URL.
3. The application fetches data from the JSONPlaceholder API.
4. The retrieved data is displayed on the page dynamically.

4. Explanation of Data Flow

1. The user interacts with the navigation bar to visit different pages.
2. React Router dynamically loads the corresponding component based on the route.
3. The Post component fetches data from an external API when a dynamic post URL is visited.
4. If an invalid route is accessed, the NotFound component displays an error message.
5. Data flows between user actions, routing, component rendering, and API responses.

5. Conclusion

This document outlines the detailed data flow in a React application that utilizes dynamic routing.
The application demonstrates how React Router manages navigation, how data is dynamically fetched,
and how the UI updates based on route parameters.